

В.І. Калашніков, І.П. Хавіна

ОБРОБКА МАГНІТНО-РЕЗОНАНСНИХ ВІДКЛИКІВ

Лабораторний практикум з курсу
«Спеціалізовані комп'ютерні системи»

Харків НТУ «ХПІ» 2010

УДК 004 53.082.79
ББК 32.973.2
К 17

Рецензент:

В.В. Скороделов, канд. техн. наук, проф. НТУ «ХП»

Калашніков В.І.

К 17 **Обробка магнітно-резонансних відкликів** : лабораторний практикум
В.І. Калашніков, І.П. Хавіна. – Харків : НТУ «ХП», 2010. – 99 с.

ISBN

Лабораторний практикум присвячений вивченню технічних та програмних принципів побудови апаратного та програмного забезпечення томографічних систем, які засновані на вимірюванні ядерно-магнітно-резонансних явищ.

У лабораторних роботах використовуються пакети Micro-Cap v.7-9 та Derive v. 6, які дозволяють виконувати моделювання електронних схем та математичних моделей фізичних об'єктів, реалізовувати спеціальні обчислювальні процедури обробки масивів даних та їхню візуалізацією, розраховувати конфігурацію магнітних полів та ін.

Призначено для студентів спеціальності «Спеціалізовані комп'ютерні системи».

Іл. 55. Бібліогр. 5 назв.

ББК 32.973.2

ISBN

© В.І. Калашніков, І.П.
Хавіна, 2010
© НТУ «ХП»

ЗМІСТ

| | |
|--|-----------|
| Вступ | 5 |
| 1. Моделювання в середовищі Micro-Cap | 6 |
| 1.1. Основні функціональні характеристики пакета | 6 |
| 1.2. Послідовність побудови схем | 6 |
| 1.3. Числовий аналіз схеми..... | 8 |
| 2. Програмування в пакеті DERIVE..... | 13 |
| 2.1. Значення операндів | 13 |
| 2.2. Змінні та оператори | 13 |
| 2.3. Вбудовані циклічні оператори | 15 |
| 2.4. Графіка пакета DERIVE..... | 29 |
| 3. Лабораторна робота № 1. Моделювання блоків прийому та формування сигналів ядерно-магнітного резонансу | 34 |
| 3.1. Мета роботи | 34 |
| 3.2. Виконання роботи | 34 |
| 3.3. Обчислення зсувів за результатами обробки | 39 |
| 3.4. Завершення роботи..... | 39 |
| 4. Лабораторна робота №2. Перетворення Радону та метод зворотного проектування..... | 40 |
| 4.1. Мета роботи | 40 |
| 4.2. Виконання роботи | 40 |
| 4.3. Висновки | 46 |
| 4.4. Порядок виконання роботи | 46 |
| 5. Лабораторна робота №3. Обробка зображень і їх гістограм | 47 |
| 5.1. Мета роботи | 47 |
| 5.2. Виконання роботи | 47 |
| 5.3. Нелінійна обробка яскравості | 53 |
| 5.4. Виконання роботи та оформлення звіту | 56 |
| 6. Лабораторна робота № 4. Розрахунки та дослідження магнітних полів для стимуляції електромагнітного резонансу атомних ядер..... | 57 |
| 6.1. Мета роботи | 57 |
| 6.2. Порядок виконання роботи | 57 |
| 6.3. Приклад проведення лабораторної роботи..... | 57 |
| 6.4. Вибір варіанта роботи | 59 |

| | |
|--|-----------|
| 6.5. Результати дослідження..... | 67 |
| 6.6. Оформлення звіту..... | 67 |
| 7. Лабораторна робота №5. Моделювання рівнянь Блоха..... | 68 |
| 7.1. Мета роботи | 68 |
| 7.2. Теоретичні відомості..... | 68 |
| 7.3. Порядок виконання роботи | 70 |
| 7.4. Завдання на проведення досліджень | 74 |
| 8. Лабораторна робота №6. Лінійна фільтрація зображень методом згортки | 75 |
| 8.1. Мета роботи | 75 |
| 8.2. Теоретичні відомості..... | 75 |
| 8.3. Виконання роботи | 78 |
| 8.4. Хід виконання роботи | 82 |
| 9. Лабораторна робота №7. Нелінійна фільтрація зображень | 83 |
| 9.1. Мета роботи | 83 |
| 9.2. Теоретичні відомості..... | 83 |
| 9.3. Види нелінійних фільтрів з вікном 3×3 | 84 |
| 9.4. Вихідне зображення та зашумлення..... | 87 |
| 9.5. Побудова програми нелінійної фільтрації..... | 88 |
| 9.6. Сигма фільтри..... | 91 |
| 9.7. Хід виконання роботи | 93 |
| 10. Лабораторна робота № 8. Розкладання дискретних випадкових векторів у ряд Карунена-Лоева за власними векторами кореляційної матриці..... | 94 |
| 10.1. Мета роботи | 94 |
| 10.2. Теоретичні відомості..... | 94 |
| 10.3. Послідовність виконання роботи..... | 94 |
| 10.4. Приклад виконання лабораторної роботи | 95 |
| 10.5.Звіт та висновки..... | 98 |
| Список літератури..... | 99 |

Вступ

Лабораторний практикум з дисципліни «Спеціалізовані комп'ютерні системи» спрямований на системне вивчення та засвоєння технічних і програмних принципів побудови та компонування апаратного і програмного забезпечення складних вимірювальних систем на прикладі томографічних систем, у яких використовуються збудження та вимір ядерно-магнітно-резонансних явищ з апаратною обробкою оцифрованих даних. Накопичені дані в таких системах проходять складну комп'ютерну обробку з метою візуалізації внутрішньої структури досліджуваного об'єкта.

У розділі 1 надано моделюючий пакет Micro-Cap v. 7–9, а у розділі 2 математичний пакет Derive v. 6, які дозволяють виконувати професійне моделювання електронних схем та математичних моделей фізичних об'єктів, опрацьовувати спеціальні обчислювальні процедури обробки масивів даних з їхньою подальшою візуалізацією, розраховувати конфігурацію магнітних полів та ін. Пакети вільно розповсюджуються фірмами-розробниками через Інтернет. Перевага математичному пакету Derive v.6 надається тому, що він, на відміну від інших професійних математичних пакетів, є пакетом символної математики, у якому на базі невеликого числа операторів можна створювати складні функції математичного перетворення та спрощення формульних виразів.

Розділи 3-10 описують вісім лабораторних робіт.

Звіти з лабораторних робіт повинні оформлюватися без викладу теоретичних положень, але з аналізом і обґрунтуванням отриманих результатів кожного дослідження або розрахунку.

1. Моделювання в середовищі Micro-Cap

1.1. Основні функціональні характеристики пакета

Micro-Cap це багатосторінковий графічний редактор принципів схем де реалізована підтримка взаємодій в ієрархічній структурі процедурно відособлених і графічно позначених принципів схем.

Редактор дозволяє спільне моделювання аналогових і цифрових схем із широким набором електричних, радіотехнічних і логічних компонентів, моделювання схем з компонентами й сигналами, які представлені функціями z - і s -перетвореннями функції дійсної змінної, графічне і табличне відтворення результатів моделювання з можливістю їх всебічної деталізації та обробки, багатоваріантний та статистичний аналіз схем, макровизначення блоків схем і пов'язування їх з типовими або спеціально створеними графічними позначеннями.

1.2. Послідовність побудови схем

1.2.1. Створення нового файлу для схеми моделювання

У пункті **File** головного меню вибирається підпункт **New**, а в ньому – рядок **Schematic**. На екрані монітора відкривається чисте поле графічного редактора зі смугою робочих кнопок-вказівників угорі для оперативного переключення режимів роботи пакета. Перелік доступних компонентів і макровизначень міститься в пункті меню **Component**. Усі графічні елементи, виведені на поле креслення, прив'язуються до вузлів сітки із комітками 2,5 мм (0.1 дюйма). Максимальний розмір поля креслення не обмежується. Ліфти прокручування та багато кнопок виконують свою загальновідому роль. Компоненти на полі креслення переміщуються натиснутою лівою кнопкою миші та повертаються клацаннями по правою кнопкою.

Після установлення потрібного компонента на необхідне місце спливає вікно установки її параметрів. Числові значення записуються в основних одиницях вимірювання фізичних і відносних величин у загальновизнаній формі. Допускається запис значень похідних від основних одиниць шляхом приписування відповідної літери безпосередньо за молодшим розрядом дробового числа:

| | | | | | | | | | |
|-------|-------|------|-------|-------|------|------|------|------|------|
| Femto | Pico | Nano | Micro | Milli | unit | Kilo | Mega | Giga | Tera |
| F, f | P, p | N, n | U, u | M, m | - | K, k | MEG | G, g | T, t |
| 1e-15 | 1e-12 | 1e-9 | 1e-6 | 1e-3 | 1 | 1e3 | 1e6 | 1e9 | 1e12 |

Тригерні квадратні кнопки-віконця з написами **Display** і **Display Pin Name** дозволяють вивід відповідного тексту на поле креслення.

Зображення елемента разом з його текстовим обрамленням або роздільно можна пересувати по полю креслення натиснутою лівою клавішею миші. Подвійне натиснення на зображення елемента або на його текстовий фрагмент відкриває вікно для установки раніше введених значень і установок.

1.2.2. Видалення, виділення і копіювання фрагментів схеми

Видалення будь-якого фрагмента схеми, виділеного окресленням його натиснутою лівою клавішею, здійснюється клавішею **Delete**. Виділене зображення копіюється у власний буфер пакета натисканням клавіш **Ctrl** і **C**. Виведення останнього в указане місце виконується по натисканню **Ctrl** і **V**.

1.2.3. Приховані з'єднання та текстові позначення на схемі

При побудові складних схем зручно використовувати сполучні елементи **Tie** з однаковими іменами. Всім вузловим з'єднанням у схемах автоматично присвоюються номери. Номери є мітками, візуалізацію яких здійснюють натисканням на режимну кнопку із зображенням одинички. Можна також ввести власні мітки або цілі текстові повідомлення, якщо натиснути режимну кнопку із символом **T** і вказати місце її розташування лівою клавішею миші. Текстові записи на полі екрана або сполучних лініях схеми можна візуалізувати або сховати кнопкою з написом **TEXT**.

1.2.4. Команди керування

На полі креслення можна розташовувати текстові команди керування. Відмінність команд від інших текстів полягає в наявності точки перед іменем керуючого оператора. Найбільш значущими для початкового використання моделюючого пакета є команди.

.DEFINE txt1 txt2 оператор заміни тексту1 на текст2 у будь-якому атрибуті схемного елемента, крім атрибута **PART**:

.DEFINE v1 t*sin(t)

.PARAM 'список' присвоює змінним списку значення:

.PARAM r=3, h= -7 або тексти:

.PARAM r={t*sin(t) +pow(t,2)}, h={t*t+ cos(2*PI*t)}

.IC 'список' змінним списку надаються початкові значення на увесь час чисельного аналізу (вищий пріоритет) **.IC V(5)=1 V(2)=.45**

.NODESET 'список' Змінним списку надаються початкові значення лише на початок чисельного аналізу (нижчий пріоритет) **.NODESET V(5)=1 V(2)=0.45**.

1.3. Числовий аналіз схеми

1.3.1. Види числових аналізів

Режим чисельного аналізу вибирається в пункті головного меню **Analysis**, де представлено три основні види чисельних розрахунків:

Transient – розрахунки перехідного процесу в часі;

AC – розрахунки частотних характеристик на змінному струмі;

DC – розрахунки коефіцієнта передачі на постійному струмі.

1.3.2. Установка параметрів для Transient Analysis

Числові значення записуються в основних або похідних одиницях.

а) Параметри інтегрування

Time Range – завдання тривалості ($t_{max}[t_{min}]$), де $t_{max} > 0$, $t_{min} \geq 0$;

Maximum Time Step – завдання максимального кроку інтегрування: при відсутності значення крок рівний $(t_{max} - t_{min})/50$. Якщо на черговому кроці обчислення припустима відносна погрішність (0.001) не отримана, крок автоматично зменшується до 2^7 разів;

Number of Points – кількість значень ординат, що записуються у текстовий файл *.TNO із кроком Δt по осі часу

$$n = (t_{max} - t_{min}) / \Delta t + 1.$$

б) Параметри видачі результатів аналізу

Дванадцяти колонкова таблиця на рисунку 1.1 визначає вид розрахунків на екрані монітора. Перші шість колонок, що представлені кнопками, у порядку їх розташування визначають: вид розмітки шкал по вертикалі й горизонталі (лінійна/логарифмічна), колір кривої, запис у текстовий файл *.TNO, запам'ятовування кривої перехідного процесу у файл *.USR, який можна використовувати як джерело сигналу типу **User Source** і розрахунки функції розподілу ймовірності виведеного процесу.

Строчки таблиці з однаковими цифрами в колонці **P** зображують відповідну групу кривих на окремому графічному полі.

У колонках **X** і **Y** вписують алгебраїчні вирази, за якими обчислюються значення абсциси і ординати кривої, виведеної на графічне поле. Для незалежної змінної звичайно використовують простий вираз: час **T** або **t**, електричну напругу між мітками вузлів **V(7,2)** або напругу на схемному елементі **V(R5)**, **v(Out)** або електричний струм через схемний елемент **i(7,2)**, **I(R1)**. Загальний дріт (земля) має нульовий номер (0). Клема вольтметра ”+”

завжди підключається до точки, яка стоїть у виразі на першому місці. Струм, що входить у цю точку, позитивний. Якщо в дужках зазначене ім'я схемного елемента, то полярність клем вимірника, підключеного до схемного компонента, і полярність висновків останнього збігаються. Можливо задавати складні математичні перетворення за участю схемних змінних, наприклад, $\text{SQRT}(\text{ABS}(v(\text{Ubx}))) * v(\text{Ubx}) * t$. Змінними у виразах можуть бути напруги та токи в елементах схеми або між відзначеними на схемі точками.

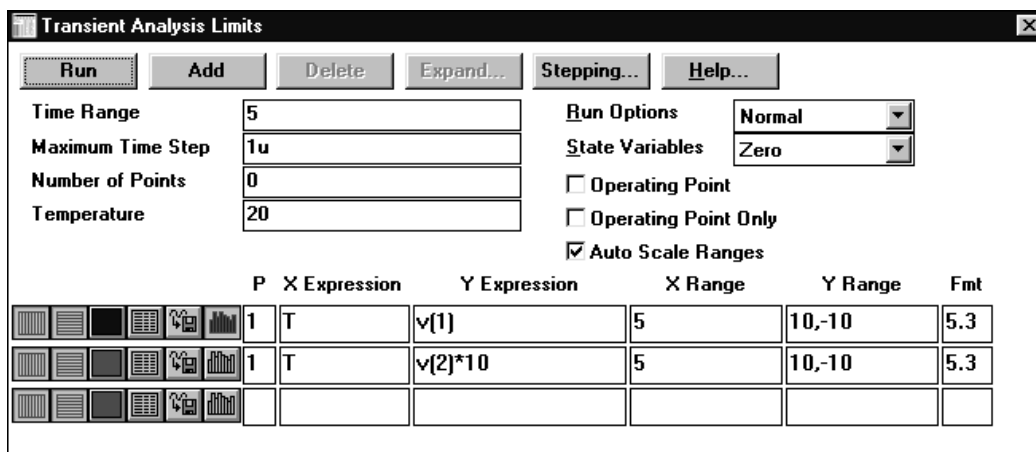


Рисунок. 1.1 – Вікно установки параметрів розрахунків перехідних процесів

У колонки **X Range** і **Y Range** вписуються граничні значення для шкал у вигляді ($X_{\max}, X_{\min}, \Delta X$) і ($Y_{\max}, Y_{\min}, \Delta Y$).

Fmt – формат числових значень: цифра перед десятковою точкою показує кількість розрядів у цілій частині, після точки – кількість дробових розрядів.

в) *Режими змінних стану:*

Run Option (Normal, Save, Retrieve) – установка режимів виконання:

процес відтворюється і не зберігається при виході зі стану аналізу;

процес зберігається у файлі *.TSA ;

процес відтворюється з файлу.

State Variables – визначення джерела початкових значень змінних стану при черговому пуску (**Run**) програми аналізу:

Zero – для всіх змінних стану встановлюються нульові значення.

Read – початкові значення прочитуються з файлу *.TOP. Цей файл створюється у редакторі змінних стану (**State Variables Editor**) головного меню, що перебуває в пункті, **Transient**, після натискання кнопки **Write**. У зазначений файл при цьому переписуються значення змінних стану, отриманих на момент зупинки розрахунків.

Leave – як початкові значення використовуються кінцеві значення

змінних стану, отримані в попередньому сеансі розрахунків. При цьому перший сеанс виконується з нульовими початковими умовами.

Operating Point – обчислюються результуючі значення (робочі точки) змінних стану за постійним струмом у всіх вузлових точках схеми. Візуалізація цих значень здійснюється після виходу з аналізу та натискання режимної кнопки із зображенням у вузлових точках квадрата із числом 13.

Operating Point Only – те ж саме, що й у попередньому режимі, але без обчислення перехідних процесів. Розрахунки не виконуються, якщо матриця системи диференціальних рівнянь особлива (сингулярна) або вироджена, про що робиться попередження.

Auto Scale Ranges – устанавлюється режим автоматичного визначення границь для шкал. Після відключення цього режиму границі шкал можна відредагувати та використовувати у подальших сеансах аналізу.

2) Команди керування аналізом

Run – запускає програму чисельних розрахунків;

Add – додає рядок у таблицю нижче тієї, у якій перебуває курсор;

Delete – видаляє рядок, у якому розташований курсор;

Stepping – устанавлює режим багатокрокового виконання, шляхом циклічної зміни числового параметра якого-небудь із атрибутів схемного елемента.

1.3.3. Установка параметрів для AC Analysis

У цьому розділі аналізу розраховуються частотні та шумові характеристики досліджуваної схеми. Призначення кнопок, що залишилися, у таблиці параметрів колишнє.

Frequency Range – установка значень шкали частот F_{\max} , F_{\min} .

Number of Points – число точок n у таблиці оцифровування частотних графіків залежить від виду розмітки шкали. Для лінійної та логарифмічної розміток інтервал між точками обчислюється відповідно до формул:

$$\frac{F_{\max} - F_{\min}}{n - 1} \quad \text{і} \quad (n-1) \sqrt{\frac{F_{\max}}{F_{\min}}}.$$

Якщо необхідно обчислити величину шуму, породженого яким-небудь схемним компонентом, то заповнюються наступні два віконця. А якщо ні, то їх залишають без уваги.

Noise Input – у цьому полі записується ім'я схемного джерела, у якому обчислюється величина вхідного шуму. Залежно від типу шумливого компонента може враховуватися і розраховуватися сума шумів трьох видів:

теплого, дробового та мерехтливого. Одиниця виміру шуму $V/\sqrt{Гц}$.

Noise Output – у цьому полі у форматі вузол1[,вузол2] записуються номери вузлів схеми, між якими (або для якого) має бути обчислений шум. Обчислення виконується при натисканні на п'яту кнопку.

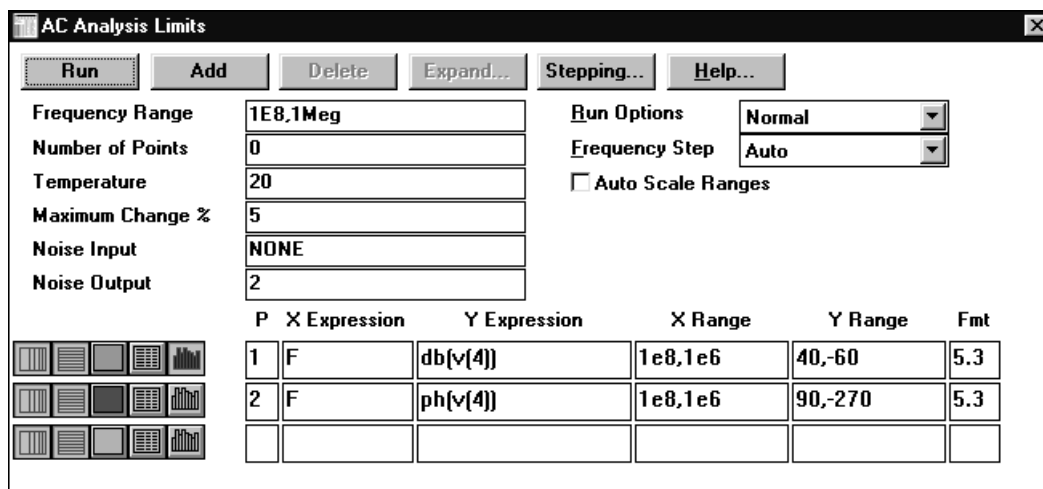


Рисунок 1.2 – Вікно установки параметрів частотного аналізу

Перші дві кнопки керування виглядом графіків задають логарифмічну або рівномірну розмітку вертикальної та горизонтальної осей. У свою чергу, цифрування осей графіків управляється спеціальними операторами, записуваними в колонках X,Y Expression, наприклад:

$db(v(i))$ – амплітуда напруги в i -тому вузлі: $20(\lg(v(i)))$ [дБ];

$ph(v(i))$ – фазове зрушення напруги i -того вузла схеми в градусах;

$gd(v(i)) - \frac{d\varphi}{d\omega} = \frac{1}{2\pi} \cdot \frac{d\varphi}{df} \equiv \frac{1}{360^\circ} \cdot \frac{\Delta(ph(v(i)))}{\Delta(F)} \left[\frac{\text{рад}}{\text{Гц}} \right]$ інтенсивність фазового

зсуву та ряд інших.

Інші поля вікна установки параметрів частотного аналізу ідентичні аналогічним полям у вікні установки параметрів аналізу перехідних процесів.

1.3.4. Установка параметрів для DC Analysis

У цьому аналізі обчислюються статичні залежності вихідної величини від вхідної (рис. 1.3). Джерелами вхідних впливів на заданий вузол схеми мають бути вбудовані джерела постійного струму або напруги, що мають ім'я атрибута частини (Path), що починається на букву I або V. Це джерела з розділу **Waveform sources** пункту головного меню **Component**.

У поле **Input 1** або **Input 1** і **Input 2** (для схем із двома входами) містяться імена вхідних джерел. У полях **Input 1 Range** і **Input 2 Range** у форматі **Max,Min,Step** вказуються граничні значення вхідних впливів і крок їх

змін.

Якщо одне джерело зазначене в колонку X Expression, як змінна осі абсцис, наприклад, V(Vin2), то друге джерело Vin1 подається на графіку разом кривих.

| P | X Expression | Y Expression | X Range | Y Range | Fmt |
|---|--------------|--------------|---------|---------|-----|
| 1 | v[2] | v[7] | 15,-5 | 6,-4 | 5.3 |
| | | | | | |

Рисунок 1.3 – Вікно запису параметрів розрахунків на постійному струмі

Інші поля та кнопки аналогічні попереднім режимам.

2. Програмування у пакеті DERIVE

Математичний пакет DERIVE версій 7–9 є зручним інструментом символічних математичних перетворень, по-перше, і по-друге формульним і процедурним обчислювачем. У рамках лабораторного практикуму необхідно буде будувати символічні математичні вирази, спираючись на вивчені форми запису основних операторів, процедур і правил установки типів операндів.

Щоб при програмуванні використовувати верхній і нижній регістри клавіатури необхідно після запуску пакета в підменю Option (Modesetting) вибрати Word і Sensitive.

Завдання алгебраїчних об'єктів на полі Алгебри (Основне вікно монітора) проводиться через командну стрічку. Виділення лексично й/або синтаксично закінчених конструкцій на полі Алгебри здійснюється однократним або багаторазовим клацанням миші по фрагменту у реченні, що цікавить. Виділена частина може бути оброблена (обчислена, перетворена, замінена, ...), якщо використовувати, принаймні, одну з режимних кнопок із гравировками [=] або \approx . З першою кнопкою відбудуться символічні перетворення і абсолютно точні обчислення, а з другою – наближені.

2.1. Значення операндів

Значеннями операндів у пакеті DERIVE є такі об'єкти:

- ціле число або цілий дріб $17; 65/123;$
- дійсне число $5.67 \cdot 10(-6);$
- літерал “ab37”; “_”; “ тобто будь-який символічний набір у лапках”;
- ідентифікатор Ya; v1z; g; wf_; ... ;
- вектор – список компонентів будь-яких типів, узятий у квадратні дужки, наприклад, $[a0, a1, \dots, an]$;
- матриця – вектор з векторними компонентами, наприклад: $[[e, y, Y], [g, a0, 17]]$ або $[e, y, Y; g, a0, 17]$ – є матрицею 2×3 .
- функція – ідентифікатор | ідентифікатор(список формальних параметрів);
- множина – список елементів, взятий у фігурні дужки: $\{e, y, Y\}$.

2.2. Змінні та оператори

Ідентифікатор, що починається з букви може служити іменем змінної, якщо він не є зарезервованим ключовим символом або словом (про це перед аналізом надаватися попередження). Змінній присвоюється значення за

допомогою оператора присвоювання

$\text{ім'я_змінної} := \text{значення}.$

Функція визначається будь-яким синтаксично правильним виразом теж за допомогою оператора присвоювання

$\text{ім'я_функції}(\text{список формальних параметрів}) := \text{визначення через параметри}.$

За умови виклику функції, визначеної тільки іменем, параметри у функцію передаються значеннями операндів, згаданих у визначенні функції. У функцію зі списком формальних параметрів на фіксовані визначенням місця вписуються фактичні імена або фактичні значення. Не названі в списку параметри з визначення функції передаються своїми значеннями. Функції повертають значення тих змінних, які визначалися останніми.

Привласнене операнду значення зберігається доти, поки оператором присвоювання це значення не буде замінено іншим. Протягом однієї сесії це правило діє і у тому випадку, коли з поля алгебри всі вирази присвоювання вилучені.

Оператори присвоювання можна згрупувати в програмний блок. Зовні запис блоку має вигляд вектору, компонентами якого є оператори присвоювання. Виконання такого оператора присвоювання значень змінним відбувається в умовах передічування їх у списку. Блок може бути іменований, наприклад

$L_blok := [x := 7, y := x2, z1 := f(x, y)].$

У результаті виконання цього блоку x та y послідовно одержать значення відповідно 7 і 49, після чого в визначену раніше функцію $f(x, y)$ замість x та y будуть підставлені їхні нові фактичні значення й результат обчислення функції буде привласнений змінній $z1$. Привласнені в такий спосіб значення будуть зберігатися доти, поки іншими операторами присвоювання їх значення не будуть змінені. Таким способом зручно повторно ініціювати початкові значення змінним, які в результаті роботи з пакетом могли набути інших значень.

Усі імена операндів, яким щось привласнювалося, незалежно від місця їх розташування на полі алгебри, передають свої значення в будь-які формульні й операторні вирази, де зустрічається їхнє ім'я.

Якщо курсором виділити оператор цілком або тільки який-небудь фрагмент його правої частини, то значення оператора після виконання

заміняється новим зміненим виглядом виділеного раніше фрагмента. Це зручно при покроковій перевірці запрограмованих дій і/або прискорення обчислень у випадках, коли в операторах і формулах утримується глибоке ієрархічне внесення множини функцій, викликаних за іменами.

Серед вбудованих операторів-функцій в DERIVE є оператори стандартних функцій для звичайних і комплексних аргументів, функції з векторними аргументами, циклічні й умовні оператори, оператори розв'язку рівнянь і їх систем та ін. Вбудовані в DERIVE оператори та бібліотеки утиліт дозволяють установити власні оператори, процедури, програми. Добірка процедур за заданою темою, завантажена у виконувачу задачу як утиліта, на екрані монітора не відображається, однак усі імена функцій зі списками параметрів стають доступними, як і вбудовані функції.

2.3. Вбудовані циклічні оператори

В DERIVE є принаймні 4 оператори, які реалізують циклічні обчислення, а саме: обчислення суми членів ряду, обчислення добутку членів ряду, табулювання функції та підстановка значення функції як її ж аргументу.

2.3.1. Обчислення суми членів ряду

Загальна форма запису оператора підсумовування в командному рядку:

$$\Sigma(f(k), k, kstart, kend) \quad \text{і} \quad \Sigma(Vg \downarrow k, k, kstart, kend),$$

де $f(k)$ – вирази функції від змінної циклу k ;

$Vg \downarrow k$ – k -тий елемент вектора Vg ;

$kstart, kend$ – початкове та кінцеве значення змінної циклу.

Нумерація компонентів вектора завжди починається з одиниці.

Якщо в будь-якому виразу на полі алгебри виділити текст k -го члена ряду, то можна скористатися «гарячою» кнопкою із зображенням знака $[\Sigma]$ і у вікні, що спливає, задати ім'я змінної циклу та її межі зміни. При закритті вікна вирази суми для виділеного фрагмента буде виведено на поле алгебри у вигляді нового рядка.

Будь-який виділений рядок на полі алгебри можна перетягнути нагору або вниз, зачепивши її правий верхній кут натиснутою лівою клавішею миші.

Функція $f(k)$ може містити крім змінної циклу й інші символічні компоненти. Після виконання, наприклад, виразу

$$\sum_{n=0}^2 (x + y)^n$$

буде отримаємо такий результат

$$x^2 + x \cdot (2 \cdot y + 1) + y^2 + y + 1.$$

Значення меж можуть бути отримані обчисленням з будь-яких виразів, однак їх кінцеве значення має бути тільки цілим. Дійсне число може бути перетворене в ціле вбудованим оператором FLOOR(*), □ аргументом якого може бути будь-яка функція, що повертає дійсне число

$$\text{FLOOR}(7.645) \Rightarrow 7;$$

$$\text{FLOOR}(7.645 + 1/2) \Rightarrow 8.$$

Наприклад, вираз $\sum_{n=M_{2,3}}^{\text{DIMENSION}(M)} M_n$, підсумовує рядки матриці M невідомого

розміру, а номер рядка, з якого починається підсумовування рядків, дорівнює числу, розташованому в цій же матриці на другому рядку в третьому стовпці, може бути задане в командному рядку так

$$((M \downarrow n, n, M \downarrow 2 \downarrow 3, \text{DIMENSION}(M))).$$

Тут оператор DIMENSION(M) визначає та підставляє кількість рядків у матриці M , тобто кількість компонентів у векторі M з векторними компонентами.

2.3.2. Обчислення добутку членів ряду

Обчислення добутку членів ряду у всьому схожі на обчислення суми. Відмінність лише в тому, що запис у командному рядку і зображення добутку в алгебраїчному вікні замість знака Σ (або аббревіатура SUM) містять грецьку букву верхнього регістру Π (або аббревіатура PRO). Наприклад, для обчислення добутку всіх елементів третього стовпчика матриці M , наведеної вище, у командному рядку набирають такий текст

$$((M \downarrow n \downarrow 3, n, 1, \text{DIMENSION}(M))).$$

Дозволене внесення циклічних операторів підсумовування і множення один до одного будь-якої глибини. Розгорнення цих циклів при обчисленнях починається з найглибшого. Приклад вираження із вкладеними циклами, що здійснює добуток усіх елементів матриці $M := [[1, 2, 3], [2, -1, 1], [2, 3, -2]]$, у командному рядку буде подаватиметься у такий спосіб:

$$(((M \downarrow n \downarrow m, n, 1, \text{DIMENSION}(M)), m, 1, \text{DIMENSION}(M))).$$

Можна інакше, із застосуванням оператора присвоювання, якщо деякі фрагменти повторюються багато разів, як у попередньому прикладі:

$$(((M \downarrow n \downarrow m, n, 1, j := \text{DIMENSION}(M)), m, 1, j)).$$

У результаті обчислень цих операторів отримаємо число 144.

Для цього й інших подібних випадків слід пам'ятати, що, де б не був внесений оператор присвоювання, ім'я лівої частини одержує відповідне значення й виявляється доступним при виклику цієї змінної за ім'ям в будь-яких виразах поточного сеансу доти, поки іншим оператором присвоювання це значення не буде змінено. Після виконання розглянутого прикладу змінна j отримає значення, рівне 3.

2.3.3. Табулювання функцій

Таблицю в DERIVE є вектор, тип компонентів якого визначається значеннями табульованої функції. Це вбудований оператор, загальний вид запису якого в командному рядку представляється оператором наступного виду:

$$\text{VECTOR}(y, x, x0, xk, h).$$

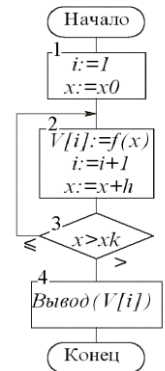
Тут перший елемент списку y є функцією аргументу x . Останній буде послідовно ухвалювати значення від $x = x0$ до $x \leq xk$ шляхом циклічного додавання до поточного значення значення кроку h . Формальний параметр y може бути представлений виразом будь-якого типу, аби тільки в нього входив аргумент x . Цим оператором проста скалярна функція $y(x)$ перетворюється в ґратчасту функцію, представлену вектором, розмірність якого є максимальним цілим числом k , що не перевищує значення:

$$k \leq \left\lceil \frac{xk - x0}{h} + 1 \right\rceil.$$

На матриці $Mx(x, y) := [[x^2, y * \text{SIN}(x)], [y, \text{EXP}(x)]]$ покажемо побудову таблиці у вигляді вектора з іменем Tabv , що має компоненти у формі матриць:

$$\text{Tabv}(y) := \text{VECTOR}(Mx(x, y), x, 1, 3, 1)$$

$$\text{Tabv}(y) := \begin{bmatrix} \begin{bmatrix} 1 & 0.01745 y \\ y & 0.3678 \end{bmatrix} & \begin{bmatrix} 4 & 0.03489 y \\ y & 0.1353 \end{bmatrix} & \begin{bmatrix} 9 & 0.05233 y \\ y & 0.04978 \end{bmatrix} \end{bmatrix}$$



Такі циклічні операції можуть проводитися за декількома змінними шляхом вкладення операторів `VECTOR(*)` один в одній. Так, наприклад, для матриці $Mx(x,y)$, після табуляції за змінною x , можна виконати оцифрування за змінною y

`TabMx := VECTOR(VECTOR(Mx(x, y), x, 1, 3, 1), y, 0.2, 0.5, 0.2).`

Результатом виконання вкладених циклічних операторів буде прямокутна матриця (2(3) з матричними елементами (2(2)

$$\text{TabMx} := \left[\begin{bmatrix} 1 & 0.00349 \\ 0.2 & 0.3678 \end{bmatrix} \begin{bmatrix} 4 & 0.006979 \\ 0.2 & 0.1353 \end{bmatrix} \begin{bmatrix} 9 & 0.01046 \\ 0.2 & 0.04978 \end{bmatrix} \right].$$

Операторному виразу із вкладеними циклами можна дати ім'я з формальним параметром x , визначивши тим самим матричну функцію. Над такою функцією можна виконувати будь-які алгебраїчні, інтегральні та диференціальні операції, які торкнуться усіх компонентів матриці одночасно. Наприклад, вираження в командному рядку

`MM(x) := VECTOR(VECTOR(x(i/j), i, 1, 3), j, 1, 2)`

після виконання визначить матричну функцію $MM(x)$ у вигляді двовимірного масиву (2×3) такого вигляду

$$MM(x) := \begin{bmatrix} x & x^2 & x^3 \\ \sqrt{x} & x & x^{3/2} \end{bmatrix}.$$

На основі цієї матриці, для прикладу, сформуємо вектор V із двома компонентами, з яких перша буде визначеним інтегралом з межами $(0,b)$ від $MM(x)$, а друга – перша похідна за x від $MM(x)$.

У командному рядку для цього повинно бути зроблено наступний запис

`V := [INT(MM(x), x, 0, b), DIF(MM(x), x)],`

який покаже на полі алгебри рядок у вигляді

$$V := \left[\int_0^b MM(x) dx, \quad \frac{d}{dx} MM(x) \right].$$

Після виконання останнього оператора буде виведений вектор

$$V := \left[\begin{bmatrix} \frac{b^2}{2} & \frac{b^3}{3} & \frac{b^4}{4} \\ \frac{2b^{3/2}}{3} & \frac{b^2}{2} & \frac{2b^{5/2}}{5} \end{bmatrix}, \begin{bmatrix} 1 & 2x & 3x^2 \\ \frac{1}{2\sqrt{x}} & 1 & \frac{3\sqrt{x}}{2} \end{bmatrix} \right]$$

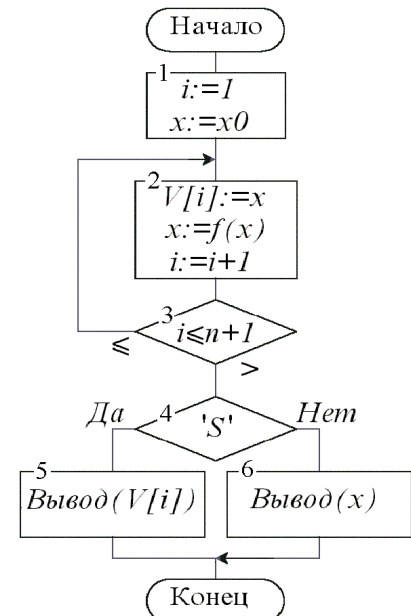
2.3.4. Ітеративний цикл

Ітеративний, рекурентний (або повторний) процес полягає в підстановці значення функції замість значення свого аргументу. Вбудованими є два оператори, що відрізняються наявністю або відсутністю однієї букви *S* наприкінці імені оператора, це

ITERATES(*f*(*x*), *x*, *x*₀, *n*);

ITERATE(*f*(*x*), *x*, *x*₀, *n*).

При наявності наприкінці імені букви *S* результат формується у вигляді (*n*+1)-мірного вектора, що містить усі результати проміжних обчислень функції *f*(*x*). Першим елементом цього вектора служить початкове значення аргументу *x*₀. При відсутності *S* результатом виконання оператора є значення функції після *n*-ної підстановки. Схема алгоритму подається на рисунку праворуч.



Ітерації застосовуються в методах чисельного розв'язання рівнянь. Найбільш популярний метод Ньютона для пошуку нулів алгебраїчного рівняння виду $f(x) = 0$. Формула ітераційного (послідовного) наближення до кореня x^* від заданого наближеного значення x_0 має такий вигляд

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad \text{або} \quad x_{i+1} = F(x_i).$$

Для прикладу за допомогою ітераційного методу Ньютона знайдемо корінь наступного рівняння $x^3 - 8x^2 - 5x + 84 = 0$.

Відповідно до формули Ньютона складемо функцію для ітераційного

наближення

$$F(x) := x - \frac{x^3 - 8x^2 - 5x + 84}{\frac{d}{dx}(x^3 - 8x^2 - 5x + 84)}.$$

Після виконання цього виразу функція отримає спрощене значення

$$F(x) := \frac{2(x^3 - 4x^2 - 42)}{3x^2 - 16x - 5}.$$

Області розташування необхідних нулів функції $F(x)$ можна одержати, побудувавши засобами пакета її графік (див. нижче). Сподіваючись знайти максимальний за величиною корінь, установимо початкове значення, що дорівнює 16, а кількість підстановок 8. Для одержання кореня із заданою точністю, тобто, щоб виводити числа із заданою кількістю десяткових розрядів, слід заздалегідь у відповідних опціях зробити необхідні установки. У розглянутому випадку точність визначено чотирма десятковими розрядами

ITERATES(F(x), x, 16, 8)

[16, 11.95, 9.418, 7.944, 7.232, 7.020, 7.000, 7.000, 7].

Функція $f(x)$ і її аргумент x можуть бути й векторами. Наприклад, для одержання ряду Фібоначі, який визначається рекурентною формулою

$$y_{i+2} = y_{i+1} + y_i, \quad \text{для } y_0 = 0 \quad \text{й} \quad y_1 = 1,$$

у командному рядку необхідно записати наступне вираження

fib(i) := ITERATES([v↓2, v↓1 + v↓2], v, [0, 1], i).

Тут вектор $\text{fib}(i) := v$ представлений двовимірними відрізками ряду (y_i, y_{i+1}) , початкове значення якого прийнято рівним $(y_0, y_1) = (0, 1)$. Векторна функція v переміщає свій другий елемент на місце першого елемента, а другий елемент заміняє сумою першого й другого. Викликавши цей оператор по імені $\text{fib}(5)$, одержимо вектор із шістьма векторними компонентами, який надано праворуч.

| | |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 5 |
| 5 | 8 |

Перший стовпець такої таблиці можна елемент за елементом вибрати за допомогою оператора VECTOR(*) і подати у вигляді вектора. Для цього через командний рядок необхідно ввести такі два рядки. Другий рядок є викликом функції по імені з конкретним значенням параметра:

$$\text{Sfib}(n) := \text{VECTOR}((\text{fib}(n)) \downarrow k \downarrow 1, k, 1, n + 1) \quad \text{Sfib}(10) \\ [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55].$$

2.3.5. Використання утиліт в ітеративних циклах

Утилітами в DERIVE називають набори процедур, підібраних на задану тему обчислень, яку можна завантажити у пам'ять поточної сесії не виводячи їх на видиме поле Алгебри. При цьому всі оператори завантаженої утиліти можуть викликатися за своїми іменами, як функції з формальними параметрами. Для кожної утиліти імена їх функцій з описом типу кожного параметра можна знайти в меню допомоги.

Приклад підстановки векторного аргументу в оператор ITERATES(*) з використанням операторів з утиліт наведено на прикладі інтегрування системи лінійних диференціальних рівнянь першого порядку методом Рунге-Куты. Процедура Рунге-Куты розташована в утиліті з іменем ODE_APPR.mth чисельне розв'язання звичайних диференціальних рівнянь. Завантаження утиліт здійснюється з головного меню $\text{File} \Rightarrow \text{Load} \Rightarrow \langle \text{ім'я утиліти} \rangle$.

Формат оператора, що викликає процедуру Рунге-Куты, має такий вигляд: RK(u, v, v0, h, k). Формальні параметри процедури включають:

- n -вимірний вектор системи рівнянь u , елементами якого є праві частини рівнянь системи;
- $(n+1)$ - вимірний вектор імен шуканих змінних v , першим елементом якого є ім'я незалежної змінної;
- $(n+1)$ - вимірний вектор початкових значень змінних $v0$;
- h – крок по незалежній змінній;
- k – число кроків.

В описуваному нижче прикладі тексту програми чисельного інтегрування системи диференціальних рівнянь закладено розв'язання протягом половини секунди наступної системи рівнянь:

$$\begin{cases} \frac{dx_1}{dt} = 2\pi x_2, & x_1(0) = 1, \\ \frac{dx_2}{dt} = -2\pi x_1, & x_2(0) = 0, \end{cases} \quad t \in [0, 0.5) .$$

Аналітичним розв'язанням цієї системи є функції $x_1(t) = \cos 2\pi t$ та $x_2(t) = \sin 2\pi t$. Чисельний метод Рунге-Куты має погрішність, пропорційну четвертому степеню кроку згідно з незалежною змінною. При кроці $h = 0.1$ помилка буде мати порядок $O(h^4) \approx 0.1^4$. Щоб підвищити точність результату

розв'язання на порядок необхідно зменшити крок в 10 разів. При цьому для перегляду всього заданого інтервалу в одну секунду слід виводити в 10 раз більше векторів розв'язання, тому що число точок на осі часу зростатиме в ті ж 10 разів, особливого змісту в яких немає.

Ідея висновку необхідного числа векторів розв'язання в необхідні моменти часу, наприклад, через інтервали, що дорівнюють десятикратно збільшеному кроку, - необхідно після кожних $k = 10$ циклів виконання процедури $RK(*)$ вивести останній вектор напочатку, його значення присвоїти вектору початкових умов $v0$ і знову 10 разів виконати процедуру $RK(*)$. Ці дії, для охоплення заданого інтервалу розв'язання в 0,5 секунд, необхідно повторити $m = 5$ раз.

Таким чином, у тексті програми, крім завдання системи рівнянь, вектора розв'язання і його початкового значення, має бути оператор із двома вкладеними циклами. Один (внутрішній), що відповідає власне процедурі Рунге-Кута і повторюється k разів, і зовнішній (ітераційний) – для повторної підстановки m разів чергового вектора початкових значень і виводу проміжних підстановок початкових векторів.

Текст програми з послідовністю операторів, записаних у текстовому форматі командного рядка, містить 5 рядків:

```
[Casemode := Sensitive, Inputmode := Word, Timesoperator := Asterisk]
[u := [2*(*x2, - 2*(*x1], v := [t, x1, x2], v0 := [0, 1, 0]]
Frk(h, k, m) := ITERATES((RK(u, v, in, h, k))↓(k + 1), in, v0, m)
F(h, k, m) := APPEND([v], Frk(h, k, m))
F(0.01, 10, 5)
```

Вектор першого рядка встановлює чутливість до регістру клавіатури, до вигляду ідентифікаторів змінних і до вигляду символу операції множення.

Другий рядок присвоює векторам u , v , $v0$ відповідно значення правих частин системи рівнянь, імена незалежної й залежних змінних і їх початкові значення.

Третій рядок визначає функцію побудови таблиці векторів розв'язання для заданих точок $Frk(h, k, m)$, а четверта – розташовує над таблицею імена змінних.

Виконання функції $F(h, k, m)$ з фактичними значеннями параметрів для кроку $h = 0.01$, кратності кроків, що пропускаються, $k = 10$ і кількості виведених значень $m = 5$ видасть такий результат

| t | x1 | x2 |
|-----|---------------|---------------|
| 0 | 1 | 0 |
| 0.1 | 0.8090170388 | -0.5877851838 |
| 0.2 | 0.3090171467 | -0.9510564578 |
| 0.3 | -0.3090167610 | -0.9510565898 |
| 0.4 | -0.8090167975 | -0.5877855123 |
| 0.5 | -1 | -0.0000004075 |

З останнього рядку видно, що результат обчислення має погрішність порядку 10^{-7} . Точне значення дорівнює нулю.

2.3.6. Найважливіші вбудовані оператори для векторів та матриць

APPEND(v1, v2, ..., vk) – виконує конкатенацію векторів, перелічених у списку параметрів, тобто робить їхнє зчеплення, утворюючи один вектор сумарної розмірності. Для утворення матриці з векторів з векторними компонентами розмірність векторних компонентів у векторів, що зчеплюються, має бути однаковою. □

APPEND_COLUMNS(A, B) – зчеплення матриць A та B стовпцями, висота яких має бути однакова. Наприклад:

[A := [a, d; b, c], B := [f, g; h, k]]

C:=APPEND_COLUMNS(A, B)

C := [a, d, f, g; b, c, h, k]

[v] – перетворення вектора v с розміром n у матрицю розміром (1(n).

` – оператор транспонування (символ «штрих»).

M', M⁻¹, DET(M) – відповідно: транспонована матриця, зворотна й визначник матриці M.

v↓i, M↓i↓j – відповідно: вибірка i-того елемента з вектора v і елемента матриці M, розташованого на i-тому рядку в j-тому стовпці. Індекси можуть позначатися вираженнями, значення яких є позитивними цілими числами.

2.3.7. Векторні та матричні процедури

В утиліті **Vectormatrixfunctions.mth** є велика кількість запрограмованих операторів-процедур для перетворення векторних і матричних операндів. Після завантаження утиліти на згадку задачі ці процедури викликаються на виконання за ім'ям. У їхній список параметрів необхідно підставляти необхідні фактичні параметри. Імена самих процедур зі списком параметрів можуть також бути фактичним параметром.

Серед існуючої множини процедур можна виділити такі:

OUTER(*r*, *s*) – повертає результат у вигляді прямокутної матриці, відповідної до зовнішнього добутку двох матриць *r* і *s*. Наприклад:

$[r := [a1, a2, a3]; s := [b, c]]$

$h := \text{OUTER}(r, s)$

$h := [a1*b, a1*c; a2*b, a2*c; a3*b, a3*c]$

MINOR(*A*, *i*, *j*) – повертає копію матриці *A*, у якій вилучені *i*-й рядок та *j*-й стовпець, тобто мінор матриці *A*.

SWAP_ELEMENTS(*v*, *i*, *j*) – міняє місцями *i*-й та *j*-й елементи вектора *v*.

Для скорочення довжини записи ієрархічних структур створюваних процедур зручно замість визначених в утилітах імен операторів підставляти їхні власні оригінальні імена зі списком параметрів або без. Перевизначення імені можна виконати за допомогою операції присвоювання, наприклад:

$\text{SWE}(v, i, j) := \text{SWAP_ELEMENTS}(v, i, j)$

2.3.8. Оператори розв'язання

Оператори **SOLVE** і **SOLUTIONS** призначені для розв'язання рівностей і нерівностей, використовуючи точні алгебраїчні методи. Функція з оператором **SOLVE** повертає результат розв'язання у формі рівностей і/або нерівностей. Функція з оператором **SOLUTIONS** повертає результат розв'язання у формі вектора значень, які задовольняють рівності або нерівності. Для прикладу візьмемо рівняння і нерівності виду: $F(x) \{<|\leq|=|\geq|>\} \text{const}$.

Праворуч від операторів розв'язання показано результат їх роботи з рівняннями, сформованими із трьох наведених нижче функцій:

$[f(x) := x^3 - 3x^2 - 13x + 1, f2 := (a*b + c)/(a*c - b),$

$f3(x) := \text{ABS}(\text{ABS}(2x + 1) - 5)]$

$\text{SOLVE}(f1(x), x) \Rightarrow x = 5 \vee x = -3 \vee x = 1;$

$\text{SOLVE}(f1(x)=0, x) \Rightarrow x = 5 \vee x = -3 \vee x = 1;$

$\text{SOLVE}([f1(x)=0], x) \Rightarrow [x = -3, x = 1, x = 5];$

$\text{SOLUTIONS}(f(x), x) \Rightarrow [1, -3, 5].$

$\text{SOLVE}(f2, c) \Rightarrow c = -a*b;$

$\text{SOLVE}([f2], c) \Rightarrow [c = -a*b];$

$\text{SOLUTIONS}([f2], c) \Rightarrow [-a*b].$

$\text{SOLVE}(f3(x)=2, x) \Rightarrow x = -4 \vee x = 3 \vee x = -2 \vee x = 1;$

$\text{SOLVE}(f3(x)>2, x) \Rightarrow x < -4 \vee -2 < x < 1 \vee x > 3;$

$$\begin{aligned} \text{SOLVE}([f3(x)=2], x) &\Rightarrow [x = -4, x = -2, x = 1, x = 3]; \\ \text{SOLVE}([f3(x)>2], x) &\Rightarrow [x < -4, -2 < x < 1, x > 3]; \\ \text{SOLUTIONS}(f3(x)=2, x) &\Rightarrow [1, -2, 3, -4]. \end{aligned}$$

Для чисельного розв'язання трансцендентних рівнянь однієї змінної в список параметрів модифікованого оператора **NSOLVE(*)** додаються ще два фактичні параметри: нижній і верхній межі діапазону, у якому може перебувати його корінь. Рівняння слід представляти в неявній формі, тобто з правою частиною, що дорівнює нулю, який записувати необов'язково, тому що при підстановці функції в оператор **NSOLVE(*)** він за замовчуванням мається на увазі. Виконується оператор натисканням на кнопку \approx . Наприклад:

$$\begin{aligned} f(x) &:= (\text{EXP}(x) + 3)/(x + \text{SIN}(x)) - 5*x; \\ \text{NSOLVE}(f(x), x, 0, 2) &\Rightarrow x = 0.7279145842. \end{aligned}$$

Точність розв'язання цього рівняння може бути задана або з меню опцій або з командного рядка оператором присвоєння:

$$\begin{aligned} \text{Notationdigits} &:= 20; \\ \text{NSOLVE}(f(x), x, 0, 2) &\Rightarrow x = 0.72791458423664709195. \end{aligned}$$

Наведений спосіб установки кількості значущих десяткових розрядів у виведених числах результатів обчислень буде діяти для всіх операторів, записаних у поточній сесії, поки аналогічним образом або через меню його значення не буде змінено на інше. Такий спосіб установки дозволений для всіх параметрів пакета **DERIVE**, що ініціалізуються.

2.3.9. Розв'язання системи рівнянь

Систему лінійних рівнянь можна задати безпосередньо в командному рядку у вигляді вектора-рядка або вектора-стовпця, як записано нижче для u .

$$\begin{aligned} u &:= [5*x + 3*y - 3*z = -10; 7*x - 4*y + 3*z = 20; 3*x + 3*y - 5*z = -26]; \\ \text{SOLVE}(u, [x, y, z]) &\Rightarrow [x = 1 \wedge y = 2 \wedge z = 7]; \\ \text{SOLUTIONS}(u, [x, y, z]) &\Rightarrow [[1, 2, 7]]. \end{aligned}$$

Подання системи лінійних рівнянь у матричному вигляді також припустимо. Нехай матриця коефіцієнтів, вектор правих частин і вектор шуканих змінних послідовно визначені через командний рядок у такий спосіб:

$$\begin{aligned} A1 &:= [5, 3, -3; 7, -4, 3; 3, 3, -5]; \\ B &:= [-10, 20, -26]; \\ X &:= [x1, x2, x3]. \end{aligned}$$

Систему рівнянь, позначену іменем U , і іменовані оператори розв'язання цієї системи можна подати так:

$$\begin{aligned} U &:= A*[X]^{\sim} = [B]^{\sim}; \\ Sv &:= \text{SOLVE}(A1*[X]^{\sim} = [B]^{\sim}, X) \quad \Rightarrow \quad Sv := x1 = 1 \wedge x2 = 2 \wedge x3 = 7; \\ St &:= \text{SOLUTIONS}(A1*[X]^{\sim} = [B]^{\sim}, X) \quad \Rightarrow \quad St := [[1, 2, 7]]. \end{aligned}$$

Імена функціям уводяться для зручного звертання до результату. Якщо правих частин системи багато, то їх можна подати у вигляді матриці і скласти функцію, яка послідовно знайде розв'язання системи для кожної правої частини. Для прикладу визначимо матрицю правих частин і головну функцію так:

$$\begin{aligned} C &:= [-10, 20, -26; -100, 200, -260; 1, 2, 3; 3, 2, 1]; \\ VS &:= \text{VECTOR}(\text{SOLUTIONS}(A1*[X]^{\sim} = [C \downarrow i]^{\sim}, X), i, 1, 4); \\ VS &:= [[1, 2, 7]; [10, 20, 70]; [7/44, -12/11, -51/44]; [21/44, 8/11, 23/44]]. \end{aligned}$$

2.3.10. Декомпозиція виразів

У пакетах DERIVE вбудовані оператори розчленовування виразів (декомпозиція). До найбільш важливих з них належать оператори виділення правої $RHS(u)$ або лівої $LHS(u)$ функції, з виразу, у якому функції зв'язані знаками відношень.

Якщо $u := f(x) \otimes g(y)$, то перший оператор поверне значення $g(y)$, а другий – $f(x)$. Тут символ відношення $\otimes \in \{=, \neq, \dots, \wedge, \vee, \dots, \mathbf{OR}, \mathbf{AND}, \dots\}$.

$$\begin{aligned} hs &:= [x = 0, y = -1, z = 2]; \\ LHS(hs)^{\sim} - RHS(hs)^{\sim} &\quad \Rightarrow \quad [x, y + 1, z - 2]. \end{aligned}$$

Декомпозиція виразів, що наведені алгебраїчною сумою, здійснюється вбудованим оператором $TERMS(*)$, який повертає вектор із числом компонентів, що дорівнюють числу доданків зі своїми знаками:

$$TERMS(x^3 - 7*x^2 + 15*x - 25) \quad \Rightarrow \quad [x^3, -7*x^2, 15*x, -25].$$

Інші оператори декомпозицій наведені в розділі **Halp**, куди можна потрапити, указавши в закладці **Index** - ім'я кожного із уже згаданих вище операторів.

2.3.11. Умовні оператори та логічні вирази

Як базисні логічні оператори використовується функціонально повний набір логічних операцій: **I**, **ЧИ**, **НІ** (**AND**, **OR**, **NOT**). Значеннями логічних

змінних $\epsilon p, q \in \{true, false\}$.

Крім базису (AND, OR, NOT) використовуються відповідно оператори еквівалентності, нееквівалентності й імплікації:

| | | |
|-----|---------------|--|
| IFF | \Rightarrow | $p \text{ IFF } q = \text{NOT } p \text{ OR } q) \text{ AND } (p \text{ OR } \text{NOT } q);$ |
| XOR | \Rightarrow | $p \text{ XOR } q = \text{NOT } p \text{ AND } q) \text{ OR } (p \text{ AND } \text{NOT } q);$ |
| IMP | \Rightarrow | $p \text{ IMP } q = \text{NOT } p \text{ OR } q.$ |

Загальна форма запису умовного оператора подається в такому вигляді

IF (test, then, else, unknown),

де test – умовний вираз, що ухвалює значення з $\{true, false\}$;

then – операторний вираз, який виконується, якщо умова дійсна;

else – операторний вираз, який виконується, якщо умова неправильна;

unknown – оператор, що повертає умовний вираз, за яким неможна облічити його значення із множини $\{true, false\}$. Іноді може бути виведений символ “?”.

Як умовні виразі можуть бути алгебраїчні вирази, пов'язані знаками відношень $\{ =, \neq, <, \leq, >, \geq \}$, про які можна сказати, що вони дійсні або неправильні, і буліві вирази, обчислення яких і ухвалює ці значення.

Припустимі вкорочені форми операторів

IF (test, then).

Операторні вирази можуть бути будь-якими синтаксично та лексично правильними з погляду DERIVE. Неправильна конструкція або ігнорується, або не випускається з командного рядка, у якому курсор установлюється на тому місці, де виявлена синтаксична помилка.

До розряду умовних операторів належать віднести безперервні кусочно-ламані функції:

SIGN(x) – функція знака, що дорівнює 1 при $x > 0$ і -1 при $x < 0$;

STEP(x) – одинична функція, що дорівнює 1 при $x > 0$ і 0 при $x < 0$;

MAX(x1, x2, ..., xn) – виділення максимального значення;

MIN(x1, x2, ..., xn) – виділення мінімального значення;

CHI(a,x,b) – функція, що дорівнює 1, якщо $a < x < b$ інакше – 0.

2.3.12. Підтвердження типу функції

В DERIVE прийнято, що функція, яка має ім'я типу значення зі знаком питання наприкінці, являється предикатом, результатом перевірки якого служать істина, або хибність (*true or false*). Звертання до предикатів функції *и*

INTEGER?(u); RATIONAL?(u); EVEN?(u); ODD?(u)

повертає значення *true*, якщо результат виконання функції є відповідно цілим, дійсним, парним і непарним числом, а якщо ні, то – *false*.

Аналогічно, звертання виду

NUMBER?(u); VECTOR?(u); MATRIX?(u) ; SET?(u)

повертає значення *true*, якщо результат виконання функції є відповідно числом, вектором, матрицею, множиною, а якщо ні, то буде повернуте значення *false*.

Якщо звертання представити як

VARIABLE?(u); STRING?(u); SUM?(u); PRODUCT?(u); POWER?(u),

тоді значення *true* буде повернуто, якщо результат функції *u* виявиться, відповідно до змінної, рядком, алгебраїчною сумою, добутком, степенем, інакше – *false*.

2.3.13. Процедурне програмування

У процедурнім програмуванні використовуються кілька нових ключових слів і можливостей для побудови багаторядкових користувацьких процедур:

PROG – оператор послідовного виконання операторів, функцій та інших визначальних конструкцій, записаних як параметри і виконувані доти, поки не зустрінуться на шляху обчислювального процесу оператори **EXIT** або **RETURN**;

LOOP – оператор циклу послідовно виконує оператори, що стоять за ним, доти, поки не зустрінуться на шляху обчислювального процесу оператори **EXIT** або **RETURN**, які закінчують цикл, повертаючи при цьому останнє значення обчисленого оператора й/або змінної ЦИКЛУ;

EXIT – де б не зустрівся цей оператор, продовження обчислювального процесу почнеться з наступного за ним оператора;

RETURN(*) – оператор, що закінчує процедуру або цикл, повертаючи значення змінних, перелічені у списку аргументів (*).

У межах контекстів **PROG** і **LOOP** використовуються як внутрішні, так і зовнішні змінні, яким можуть призначатися і/або присвоюватися нові значення. Після виходу із цих процедур значення внутрішніх змінних повертаються до вихідних, а зовнішні зберігаються зміненими.

Для демонстрації процедурного програмування розглянемо програму, що визначає процедуру гамма-корекції матриці розміром (51×51) із цілими позитивними числами від нуля до 255. Результатом роботи процедури з іменем **Mgam(Misx, γ, Mnew)** має бути нова матриця з цілими числами й тим же

самим діапазоном їх представлення (див. розділ 5.3.2).

Текст процедури можна записувати в командний рядок як визначення будь-якої функції з фактичними параметрами або у вигляді багаторядкового запису в командний рядок, який ініціюється з меню послідовним зверненням:


Options \Rightarrow Display \Rightarrow Multi-line Expression Entry.


Текст, записаний шрифтом Times New Roman, можна виділити з електронного першоджерела цілим, через буфер перенести в багаторядковий командний рядок і звідти вивести на поле алгебри.

```
Mgam(Misx,  $\gamma$ , Mnew) :=  
  PROG(  
    Mo := Misx,  
    Mn := VECTOR(VECTOR(0, i, 1, 51), j, 1, 51),  
    “створення_робітника_масиву”,  
    i := 1,  
    LOOP(  
      “початок_циклу_по_i”,  
      j := 1,  
      LOOP(  
        “початок_циклу_по_j”,  
        ko := Mo↓i↓j,  
        Mn↓i↓j := FLOOR(255*(ko/255) $\gamma$ ),  
        IF(j < 51, j := j + 1, exit) ),  
        IF(i < 51, i := i + 1, exit) ),  
    RETURN Mnew := Mn,  
    exit)
```

2.4. Графіка пакета DERIVE

2.4.1. Відображення функцій однієї змінної

Побудова графіків здійснюється для виділених курсором операторів або їх фрагментів, значення яких подіються виразами, що обчислюються за однією незалежною змінною. Для входу в графічний режим необхідно натиснути режимну кнопку із зображенням одновимірного графіка . Після натискання відкривається графічне вікно із прямокутною системою координат і своїм набором режимних кнопок.


Клацнувши мишою по кнопці з аналогічним зображенням одновимірного графіка , на полі графічного вікна буде побудований графік, відповідний до виділеної функції або її виділеного фрагменту. За допомогою режимних кнопок

або установок в головних і допоміжних позиціях меню можна отримати необхідний розмір кривої, зручну розмітку осей і багато іншого.

Одержувані зображення можна передати в інші пакети, помістивши їх у буфер. У буфер можна помістити все зображення переднього вікна або його виділену курсором частину. Вибір того або іншого варіанта визначається спеціальними позиціями пункту **Edit** головного меню: **Copy Plot Window** і **Mark and Copy**.

Для висновку сімейства декількох графіків функцій за одне входження в графічний режим необхідно всі визначення виведених функцій подати в алгебраїчному вікні елементами матриці з одним стовпцем і числом рядків, відповідним до числа виведених функцій. У командному рядку цьому випадку може відповідати такий запис для одночасного висновку, наприклад, кривих трьох функцій

$[SIN(\pi \cdot x); COS(\pi \cdot x); TAN(x)].$

Якщо на поле графіка необхідно вивести зображення точок з координатами (x, y) , то на алгебраїчне поле необхідно вивести вектор або таблицю векторів з відповідними числовими значеннями. Виділити курсором миші цю таблицю і, перейшовши в графічний режим, за допомогою режимної кнопки  ввести в область графіків. Так наприклад, щоб побудувати таблицю декількох координат точок кривої тангенсу, необхідно з командного рядка вивести і виконати наступний оператор

$VECTOR([x, TAN(x)], x, -0.5, 0.5, 0.1).$

Якщо разом з безперервної кривої бажане вивести набір точок, наприклад, на цій же кривій, то оператор, який це дозволить зробити, може виглядати так:

$[SIN(\pi \cdot x), VECTOR([x, SIN(\pi \cdot x)], x, -0.5, 0.5, 0.2)].$

Після цього виконання отримаємо вектор із двома компонентами

$[SIN(\pi \cdot x), [-1/2, -1; -3/10, -\sqrt{5}/4 - 1/4; -1/10, 1/4 - \sqrt{5}/4; 1/10, \sqrt{5}/4 - 1/4; 3/10, \sqrt{5}/4 + 1/4; 1/2, 1]]$

Керування зовнішнім виглядом графіка полегшується, якщо використовувати режимні кнопки і курсорний хрестик на графічному полі. Останній переміщується в необхідне місце клавішами зі стрілками на клавіатурі або миттєво – клацанням миші необхідному місці. Цей курсор-хрестик указує на ту точку зображення, яка має з'явитися у центрі екрана монітора, якщо клацнути по режимній кнопці, де точка розміщена в центрі кнопки. Якщо клацнути по кнопці, у центрі якої знаходиться перехрестя координатної системи,

то в центр екрана монітора зміщується початок координатної системи зображуваних графіків.

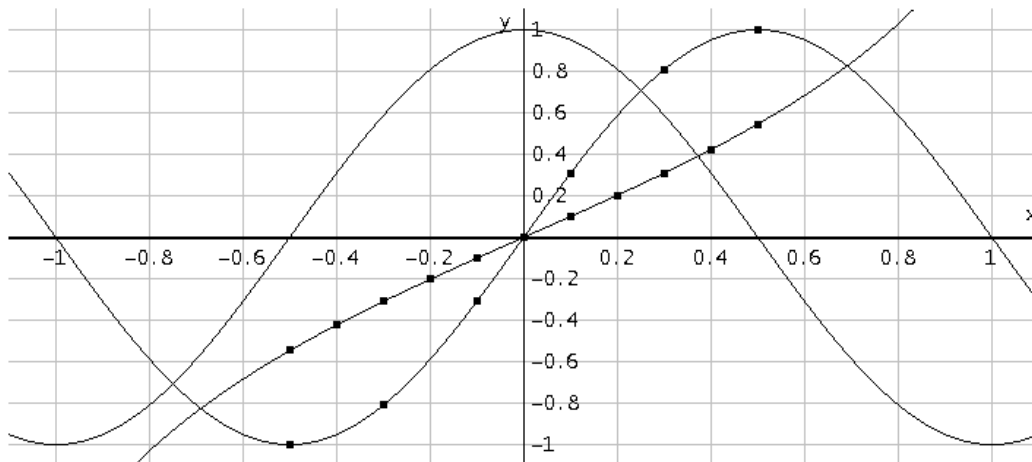


Рисунок 2.1 – Криві та крапки, побудовані за трьом розглянутими операторами


Через головне меню **Set⇒Coordinate System** можна, вибравши: **Polar**, будувати графіки й точки на графічному полі в полярній системі координат.


У пункті **Set⇒Plot Range⇒Minimum/maximum** встановлюється мінімальне (*min*) і максимальне (*max*) значення по горизонтальній і вертикальній осям і кількість (*n*) інтервалів оцифрування. Для грамотного оцифрування важливо, щоб результат обчислення ціни поділу по осях, тобто значення $c = \frac{\max - \min}{n}$ виявилося зручним для візуального зчитування числом не більш, ніж із двома значущими десятковими цифрами.


У пункті **Set⇒Aspect Ratio** можна встановити бажане відношення довжини повної горизонтальної шкали до повної вертикальної шкали, хоча можлива і візуальна зміна цього співвідношення, якщо курсором миші тягти за межі або кути переднього вікна.

Якщо по графічному полю зробити подвійне клацання мишею, то відкриється вікно дисплейних опцій, за допомогою яких можна призначити імена осям координат і задати колір їх об'єктам, вигляд розмітки поля графіків, вигляд точкових об'єктів з видами ліній, що їх з'єднують, і колір кривих.

2.4.2. Відображення функцій двох змінних

Виділене курсором визначення функції двох змінних можна подати поверхнею у тривимірній системі координат, якщо перейти у відповідний графічний режим, клацнувши мишкою по кнопці із зображенням тривимірної системи координат . Тривимірне графічне вікно теж має своє головне меню зі

своїм набором режимних кнопок. Команда для побудови поверхні надходить у момент натискання на свою кнопку із зображенням тривимірної координатної системи . Побудова чергової поверхні накладається на попереднє тривимірне зображення. Тому в пункті меню **Edit** передбачений підпункт **Delete All Plots**, по яким стираються всі попередні поверхні та криві.

Щоб зобразити в тривимірному просторі поверхню, необхідно в алгебраїчному вікні підготувати й виділити курсором відповідну функцію двох аргументів або її синтаксично закінчений фрагмент із двома незалежними змінними. Після цього натиснути кнопку виконання . У пунктах і підпунктах графічного меню утримуються опції, що дозволяють підібрати необхідний режим побудови тривимірної поверхні або кривої. Частина важливих оперативних опцій дублюється режимними кнопками.

У тривимірний простір на задане місце можна виводити точки, які описують у полі **Алгебри** таблицею тривимірних векторів з числовими значеннями координат (x, y, z), що обчислюються або задаються. Колір точок для кожної таблиці векторів встановлюється в графічному режимі з пункту меню

Insert⇒Plot⇒Color⇒Custom⇒{Top: (min=/(max), Bottom: (min=/(max))}.

У командному рядку, наприклад, таблиця для шести точок, координати яких позначені векторами, записується у такий спосіб:

[[0, 0, 0]; [1, 0, 0]; [0, 0, 1]; [0, -2, 1]; [0, -2, 0]; [1, -2, 0]].

Відрізок у тривимірному просторі задається координатами двох кінцевих точок, які утворюють матрицю із двома рядками і трьома стовпцями. Таблиця з матриць, що подає кілька окремих відрізків, у командному рядку записується так:

[[1, 0, 0; 0, 0, 1]; [1, -2, 0; 0, -2, 1]].

Тут задано два відрізки: перший – для з'єднання другої і третьої точок з наведених вище, а другий – для шостої і четвертої.

Кожна координата вектора може бути описана через загальний параметр, тоді в тривимірному просторі буде вимальовуватися просторова, параметрично задана крива. Перелічені можливості та дії з виведення на екран монітора картинок і кривих виконується для трьох видів систем координат: прямокутної, циліндричної й сферичної. Вид координатної системи встановлюється в графічному меню **Set⇒Coordinate System**. Тут же **Set⇒Plot Minimum/maximum** дає можливість задати область побудови зображення і

вибрати одиницю поділки шкали. Корисним може виявитися і установка відносних розмірів шкал за всіма трьома координатами: **Set⇒Aspect Ratio**.

Пункт меню **Options⇒Display Options** або подвійне клацання миші по графічному полю відкриває набір закладок, які визначають розташування осей, граней області, найменування і оцифровку осей. У закладці **Rotation** установлюється період обертання тривимірної картини в мілісекундах і величина приросту кута повороту від одного стану до наступного у градусах. У закладці **Color** можна встановити автоматичну зміна кольору нових картинок. Характер розцвічення всього зображення поверхні проводиться в меню **Insert⇒Plot {Parameters|color}**, коли опис зображення вже підготовлений до висновки на екран. У вікні, що відкрилося, установки параметрів зображення можна задати кількість панелей ($x \times y$), з яких буде створюватися загальна поверхня, і розцвічення від райдужної до хвилястої. Набір оперативних кнопок керування зображенням наведений на рисунку 2.2.



Рисунок 2.2 – Оперативне керування зображенням

У цьому блоці крайня ліворуч команда на побудову тривимірної картини. Чотири розбіжні та збіжні стрілки – одночасне збільшення і зменшення області побудови поверхні за осями (x , y). Такого ж типу стрілки із кружком у центрі, розташовані праворуч, змінюють масштаб усього зображення.

Равлик-Спіраль і дві горизонтально спрямовані стрілки є командами на безперервне або за кроками обертання зображення навколо осі z . Дві вертикально спрямовані стрілки покрокове повертають видиме тривимірне зображення у вертикальній площині на $\pm 90^\circ$.

Крайня права кнопка – повернення у вікно **Алгебри**.

Запам'ятовування картини й передача в інші додатки аналогічні тому, як це робиться і з двовимірними графіками. Приклад тривимірного зображення з трьома різними видами об'єктів надано на рисунку 2.3.

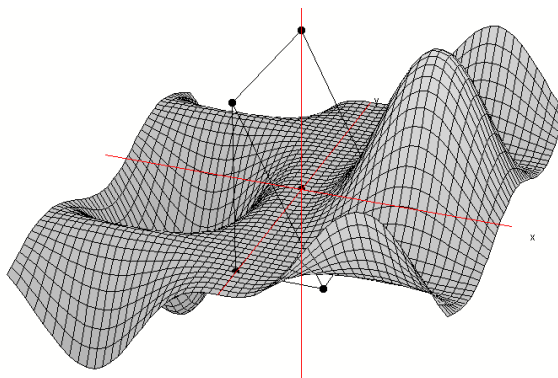


Рисунок 2.3 – Вигляд тривимірного зображення поверхонь, відрізків і крапок

3. Лабораторна робота № 1. Моделювання блоків прийому та формування сигналів ядерно-магнітного резонансу

3.1. Мета роботи

1) Одержати повне уявлення про методику побудови моделей, що відтворюють процес роботи окремих частин схемної й/або апаратної реалізації периферійних пристроїв спеціалізованих комп'ютерних систем. Як пакет моделювання використовувати Micro-Cap 7-9 (див. розділ 1).

2) Побудувати модель осциляторів, що генерують одноmodalні сигнали, які б у сумі є відгуками сигналів вільної прецесії, отриманих після імпульсного збудження зрізу зразка на частоті Лармора в зовнішньому магнітному полі з поперечним градієнтом.

3) Навчитися фіксувати в моделі динаміку процесів у зручних графічних формах із грамотним поданням шкал, їх оцифрованнями, розмітками, виділеннями і вказівками окремих особливостей на графіках кривих.

4) Навчитися використовувати обробку відтворених процесів за допомогою процедур сигнального процесора, заснованих на базовій процедурі Швидкого Перетворення Фур'є (ШПФ – англійська аббревіатура FFT).

5) Промоделювати процес перетворення частот Лармора в низькочастотну область з метою поділу відгуків окремих осциляцій за частотою і їх оцифрованнями для введення в ЕОМ.

6) Для заданого градієнта зовнішнього поля оцінити відстань осцилюючого фрагменту від ізоцентру магнітної системи.

3.2. Виконання роботи

Візьмемо за основу, щов ядрах водню досліджуваної речовини ядерно-магнітний резонанс виникає при частотах електромагнітного випромінювання накачування, рівному 10,5 мегагерц, тобто зовнішнє магнітне поле становить величину порядку 0,25 Тл. Припустимо також, що градієнт зовнішнього магнітного поля такий, що в зрізі досліджуваного зразка виникають сигнали вільної прецесії в інтервалі 100 кГц щодо середньої частоти Лармора (10,5 мегагерц). Дозвіл за частотою при заданому зрізі в інтервалі частот прецесії має скласти порядок 0,01, тобто необхідно визначати зсув частоти окремих осциляцій (гармонік) у суміші сигналів прецесії з точністю до 100 Гц.

У моделі приймання і обробки сигналів вільної прецесії мають бути осцилятори з добротністю не менш $10^5 \approx (10,5 \times 10^6 / 10^2)$. Стабільність частот ЯМР

сигналів абсолютна і визначається в основному стабільністю зовнішнього магнітного поля та наявністю локальних магнітних полів від структурних елементів досліджуваної речовини. Останні визначають так званий хімічний зсув, який необхідно враховувати при виборі конкретної частоти збуджуючого електромагнітного поля. Відмінність частот сигналів вільної прецесії від протонів, що належать різним молекулярно-ядерним структурам речовини, становить десятки-сотні герц. У моделі процесу будемо намагатися розрізняти сигнали віддалені один від одного на величину, кратну 5 кГц, у зазначеному вище діапазоні.

3.2.1. Задання даних варіанту

Кожним студентом з таблиці вибираються коефіцієнти (k_1 , k_2 , k_3) для трьох гармонік, які в спектрі відгуків частот Лармора будуть визначати їхнє положення щодо середньої частоти 10,5 МГц у такий спосіб:

$$f_i = (10,5 + k_i \times 0,005) \text{ МГц.}$$

Номер відповідно журналу групи визначається двома параметрами з таблиці, що надано нижче

$$N_{жг} = v + g \times 6,$$

де $N_{жг}$ – порядковий номер студента в журналі групи;

$v \in \{1, 2, \dots, 6\}$ – число по вертикалі;

$g \in \{0, 1, \dots, 4\}$ – число по горизонталі.

Таблиця 3.1 Задання даних варіанту

| (v,g) | 0 | | | 1 | | | 2 | | | 3 | | | 4 | | |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | k1 | k2 | k3 | k1 | k2 | k3 | k1 | k2 | k3 | k1 | k2 | k3 | k1 | k2 | k3 |
| 1 | -1 | 1 | -4 | 0 | -1 | -4 | -2 | 2 | 4 | -2 | 1 | 4 | 0 | 1 | -3 |
| 2 | 0 | -2 | -1 | 0 | -2 | -1 | 0 | 2 | -3 | 2 | -1 | -4 | 1 | 4 | -3 |
| 3 | 0 | -2 | 2 | 1 | -4 | 4 | 2 | 1 | -4 | 0 | 2 | -1 | 1 | -4 | 4 |
| 4 | 0 | -2 | -3 | -2 | -4 | -3 | 0 | -1 | -3 | -2 | 2 | 4 | -2 | 2 | -4 |
| 5 | -2 | -1 | 1 | 0 | -2 | 4 | 0 | 2 | -3 | -2 | 2 | -3 | 0 | -2 | 4 |
| 6 | 4 | -3 | 3 | -2 | -1 | -3 | 0 | -2 | -3 | -4 | -3 | 3 | 0 | 2 | 4 |

Відгуки сигналів вільної прецесії через втрату енергії загасають із сталими часу порядку декількох сотень мікросекунд. Тому при формуванні гармонійних коливань необхідно передбачити експонентний спад їх амплітуди зі сталою часу $T_1 = 0,001$ с:

$$U_i(t) = U_{mi} \cdot e^{-t/T1} \cdot \sin(2\pi f_i t + \varphi_i)$$

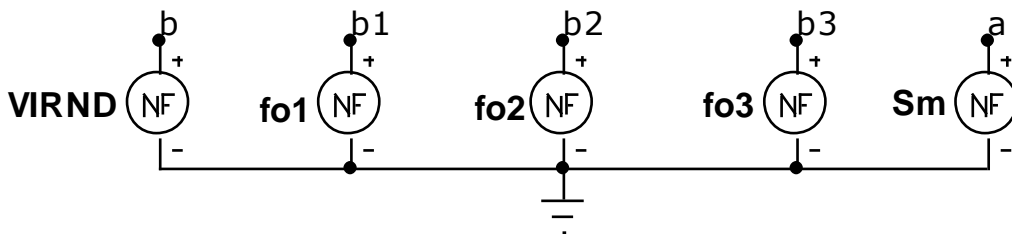
Вважатимо, що сигнал вільної індукції ми одержуємо вже після підсилювача височастотних сигналів, посиленних до рівня одиниць вольт ($U_{mi} = 1$ В), а широкосмугові шуми типу білого вп'ятеро перевищують амплітуди корисних гармонік.

3.2.2. Приклад побудови схеми та завдання параметрів сигналів

Для відтворення чистої гармоніки, спектр якої складається із синусоїдального сигналу строго однієї частоти, у моделюючому пакеті найдоцільніше вибрати блок функціонального джерела, зміна напруги або струму якої в часі задається вираженням, що обчислюється, або алгоритмічною процедурою. Розширення частотного спектра гармонік визначатиметься впливом широкосмугового шуму.

```
.define VIRND 5*RNDI(10n)
```

```
.define Sm 0.1*(V(b)+V(b1)+V(b2)+V(b3))
```



```
.define fo1 EXP(-T/0.001)*SIN(2*pi*(10.5)*10**6*T-pi/3)
```

```
.define fo2 EXP(-T/0.001)*SIN(2*pi*(10.5+0.015)*10**6*T)
```

```
.define fo3 EXP(-T/0.001)*SIN(2*pi*(10.5+0.03)*10**6*T+pi/3)
```

Рисунок 3.1 – Схема моделі, що генерує сигнал вільної індукції

На рисунку 3.1 застосовано 5 блоків функціональних джерел з іменами (ідентифікаторами):

fo1, fo2, fo3 – блоки, що генерують напруги корисних сигналів за трьома заданими коефіцієнтами;

VIRND – блок джерела, яке за допомогою процедури **RNDI(10n)** через кожні 10 наносекунд генерує прямокутний імпульс із випадковою амплітудою в межах від нуля до одного вольт.

Sm – блок, який підсумує всі названі напруги зі своїми вагами та приводить суму до діапазону одиниць вольт.

Команди **.define...** на полі рисунка 3.1 задають для обчислення конкретні

вирази, як функції часу. Для різноманітності в даному прикладі фазовий зсув кожної гармоніки в початковий момент узятий різним.

Після побудови схеми обов'язково збережіть її у файлі з Вашим ім'ям.

3.2.3. Установка параметрів аналізу перехідного процесу

Для вибору й установки параметрів режиму аналізу необхідно чітко уявляти собі всі тимчасові характеристики досліджуваного сигналу, його граничний дозвіл за часом.

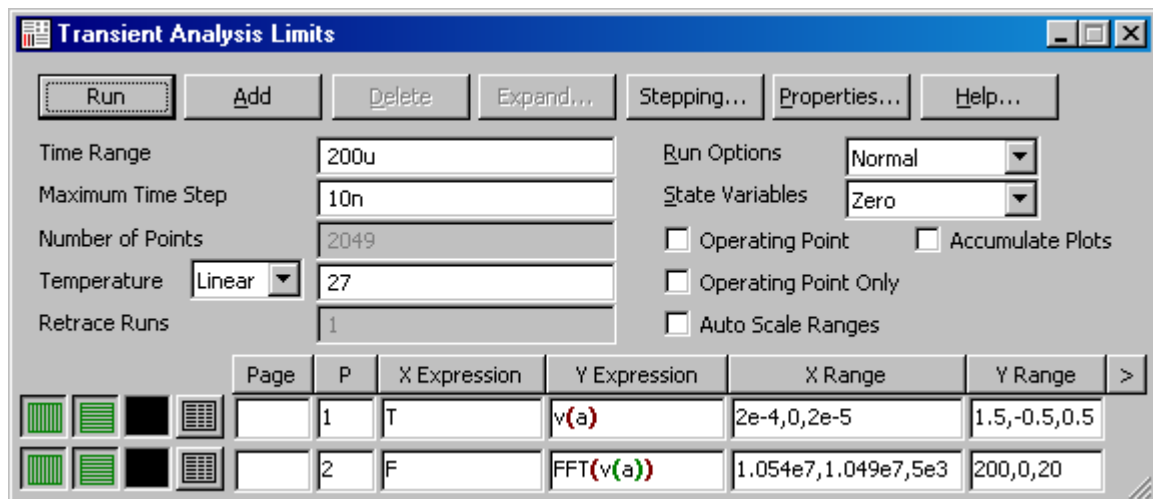


Рисунок 3.2 – Установка параметрів для процесів з рис. 3.3

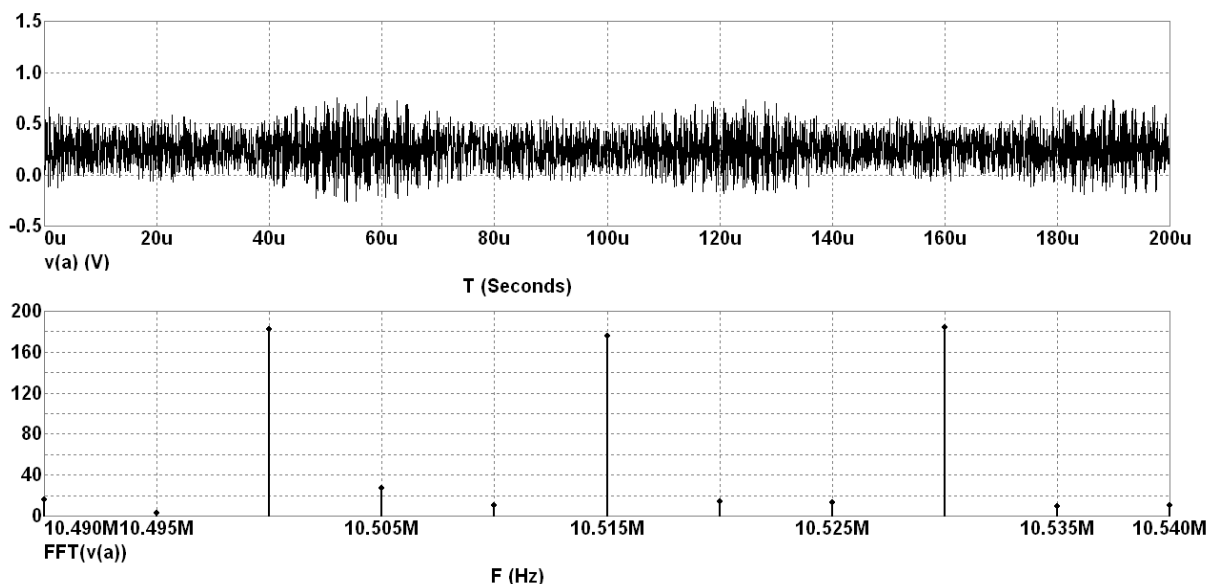


Рисунок 3.3 – Осцилограма сигналу і його спектр у діапазоні 50 кГц

При виборі установок для перехідного процесу і процедур сигнального процесора істотну роль відіграє співвідношення між інтервалом часу

спостереження перехідного процесу ΔT й частотним дозволом ΔF при обчисленні спектра процедурою швидкого перетворення Фур'є (FFT):

$$\begin{aligned} T_{max} - T_{min} &= \Delta T = \frac{1}{\Delta F}, \\ F_{max} &= N \times \Delta F, \end{aligned}$$

де T_{max}, T_{min} – відповідно максимальне й мінімальне значення часу оцифровування перехідного процесу для процедури FFT, які встановлюються у вікні, викликуваному функціональною клавішею F10 у закладці FFT;

ΔF – крок на осі частот між обчисленими і висвітленими значеннями спектра;

N – кількість значень перехідного процесу, кратна степені двійки, яка використовується для процедури FFT;

F_{max} – верхня границя частотного спектра, яка може бути відображена на осі частот графіка.

Параметри аналізу та сигнального процесора повинні бути пов'язані один з одним. Так, наприклад

$$\text{Time Range} = T_{\max},$$

$$\text{Maximum Time Step} < T_{\min},$$

$$(\text{Time Range})/(\text{Maximum Time Step}) > N$$

3.2.4. Використання функцій сигнального процесора

У розділі допомоги **Signal Processing/FFT Operators** приводиться цілий набір стандартних процедур сигнального процесора, що базуються на результатах процедури FFT.

На основі побудованої схеми випробуйте роботу деяких процедур сигнального процесора, наприклад, таких, як:

AC(u) обчислення автокореляційної функції,

IFT(S) зворотне перетворення спектра S , отриманого процедурою FFT,

REAL(S), IMAG(S) реальні і уявні частини спектра S ,

MAG(S) амплітуди спектра S , отриманого з FFT.

При включенні в роботу тієї або іншої процедури необхідно в схемі, що моделює комплексний сигнал, залишати лише деякі його складові. Зокрема, генератор шумової складової, як і інших складових, зручно видаляти не з рисунка схеми, а з оператора **.define...** шляхом множення алгебраїчного виразу

на нуль. Усі одержувані графіки подібно до рисунку 3.2 приводити у звіті з поясненням зображеного результату.

3.3. Обчислення зсувів за результатами обробки

Основна задача одержання спектра комплексного сигналу, що є відгуком вільної прецесії, полягає в тому, що за різницею частот, отриманих у спектрі гармонік і частоти накачування (у роботі це 10,5 мГц), необхідно обчислити геометричне положення конкретної гармонійної складової щодо центру магнітної системи (ізоцентру). У загальному випадку зв'язок частоти відгуку із точки, звідки цей відгук надійшов, визначається такою залежністю

$$f = \gamma_p B_0 + \gamma_p G_x x,$$

де $\gamma_p B_0 = f_0$ – [Гц] частота збудження ЯМР (частота накачування) при значенні зовнішнього магнітного поля, що дорівнює B_0 [Тл];

$G_x = \Delta B_0 / \Delta x$ – [Тл/м] градієнт зовнішнього магнітного поля B_0 ;

$\gamma_p \Delta B_0 = \Delta f$ – [Гц] зміна частоти Лармора, викликане зміною B_0 .

Враховуючи вище сказане, геометричне розташування випромінюючої точки можна обчислити за виразом

$$x_i = \frac{f_i - f_0}{\gamma_p G_x},$$

у якому слід взяти $G_x = 10$ [мТл/м]. У виконуваній роботі при такому градієнті для сигналу f03 зсув x_3 дорівнює

$$x_3 = \frac{10,53 - 10,5}{42,0 \times (10 \times 10^{-3})} = \frac{0,03}{0,42} = 0,0714286 \text{ м.}$$

3.4. Завершення роботи

У процесі виконання роботи необхідно фіксувати за всіма експериментами графіки процесів, що спостерігались. При цьому слід уважно стежити за шкалами графічних полів і їх оцифровування. У звіті за кожним графічним зображенням кривих наводити обрані значення параметрів шкал і давати інтерпретацію за формами і параметрами у кожних кривих.

4. Лабораторна робота №2. Перетворення Радону та метод зворотного проектування

4.1. Мета роботи

1) Аналітично навести в прямокутній системі координат (x, y, z) тонкий шар розподілу густини речовини. Привести дво і тривимірні графічні зображення тіней від просвічування.

2) Виконати просвічування шару паралельними променями s ($-\infty < s < \infty$) під кутами φ ($0 \leq \varphi < \pi$). За допомогою перетворення Радону в координатах (s, φ) сформуванати радонівський образ $R(s, \varphi)$. Побудувати його тривимірне графічне зображення.

3) Для заданих значень кутів φ побудувати множину тривимірних проекцій радонівських образів $Vrs(s)$ і зобразити їх графічно.

4) Виконати зворотне проектування, перетворивши кожну із проекцій у тривимірну складову радонівського образа в координатах (x, y, Vrs) і відобразити їх графічно.

5) За зворотними проекціями радонівського образа відновити двовимірну картину вихідного розподілу густини речовини в просвіченому шарі.

4.2. Виконання роботи

Ініціалізацію початкових установок математичного пакета DERIVE v.6.0 можна виконати в пунктах меню Options \Rightarrow Mode Settings... або безпосередньо скопіювати в командний рядок з даного тексту і ввести в поле Алгебри такий вектор цих установок

[Inputmode:=Word, Casemode:=Sensitive, Timesoperator:=Asterisk, Notationdigits:=4].

За наявності чотирьох розрядів чисел, виведених на екран, підставляти їх з екрана не можна. Можлива втрата точності. Необхідне внутрішнє значення задається в опціях.

4.2.1. Вибір області просвічуваного шару

Як аналітичний опис розподілу густини речовини задамо область, обмежену прямими лініями, заданими у відрізках на осях x і y : $x/a + y/b = 1$.

Область визначення подається функцією Sech(*), набравши або

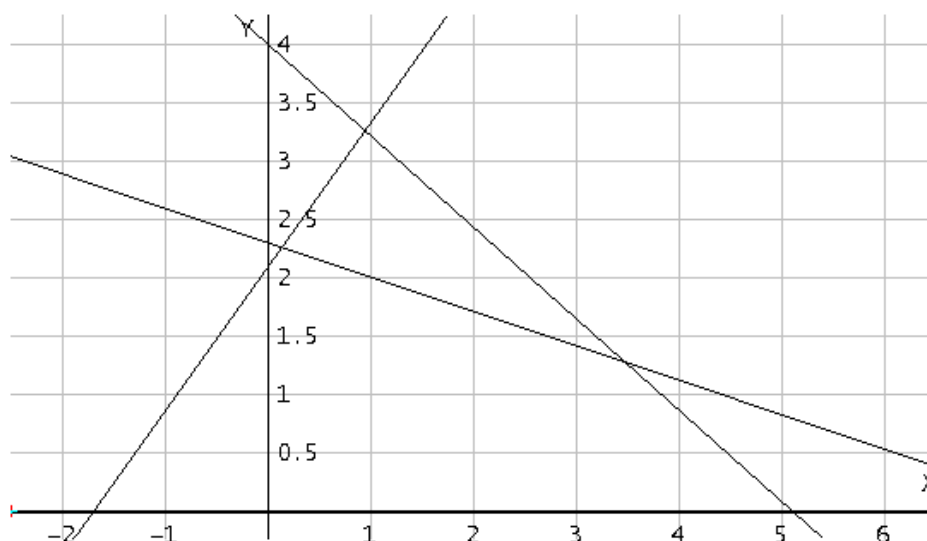
перекопіювавши в командний рядок таке вираження

$$\text{Sech}(x, y, x1, y1, x2, y2, x3, y3) := [x/x1 + y/y1 = 1, x/x2 + y/y2 = 1, x/x3 + y/y3 = 1].$$

Для побудови на площині лінії границь слід задати значення параметрам. Параметри функції $\text{Sech}(x, y, x1, y1, x2, y2, x3, y3)$ задані викладачем так, щоб трикутну область із правої сторони обмежувала перша з переліку пряма (функція вибору області $gr(x1, y1)$), з лівої сторони друга (функція вибору області $gl(x2, y2)$) і знизу – третя (функція вибору області $gd(x3, y3)$). Для контрольного прикладу виберемо такі фактичні параметри, після чого виконаємо функцію та зобразимо отримані границі на графіку (рис. 4.1).

$$\text{Sech}(x, y, 5.1, 4, -1.7, 2.1, 7.8, 2.3)$$

$$[10 \cdot x/51 + y/4 = 1, 10 \cdot x/17 - 10 \cdot y/21 = -1, 5 \cdot x/39 + 10 \cdot y/23 = 1]$$



Рисунку 4.1 – Трикутна область щільної речовини

За допомогою вбудованої функції знака

$$\text{SIGN}(u) = \begin{cases} 1, & u \geq 0, \\ -1, & u < 0 \end{cases}$$

зручно будувати функції вибору області, у якій значення функції дорівнює одиниці. Для цього як аргумент знакової функції (u) беруть неявне наведення рівняння граничної прямої, зміщують знакову функцію на одну одиницю уверх або вниз і ділять результат на 2 з необхідним знаком

$$\begin{aligned} gr(x1,y1) &:= (1 + \text{SIGN}(x/x1 + y/y1 - 1))/2; \\ gl(x2,y2) &:= (1 + \text{SIGN}(x/x2 + y/y2 - 1))/2; \\ gd(x3,y3) &:= (1 - \text{SIGN}(x/x3 + y/y3 - 1))/2. \end{aligned}$$

Враховуюче вище сказане, формулу тривимірної моделі шару одиничної товщини визначимо функцією ftr(*) таким чином

$$\begin{aligned} ftr(x, y, x1, y1, x2, y2, x3, y3) &:= \\ (1 + \text{SIGN}(0.5 - (gr(x1,y1) + gl(x2,y2) + gd(x3,y3))))/2. \end{aligned}$$

Підставивши в ftr(*) фактичні параметри, визначимо опис шару в прямокутній системі координат у вигляді функції двох змінних f(x,y), потім одержимо її арифметичний вираз і побудуємо тривимірне зображення, подане на рисунку 4.2

$$\begin{aligned} f(x,y) &:= ftr(x, y, 5.1, 4, -1.7, 2.1, 7.8, 2.3) \\ f(x, y) &:= \text{SIGN}(\text{SIGN}(210*x - 17*(10*y - 21)) + \\ &\text{SIGN}(115*x + 39*(10*y - 23)) - \text{SIGN}(40*x + 51*(y - 4)) - 2)/2 + 1/2. \end{aligned}$$

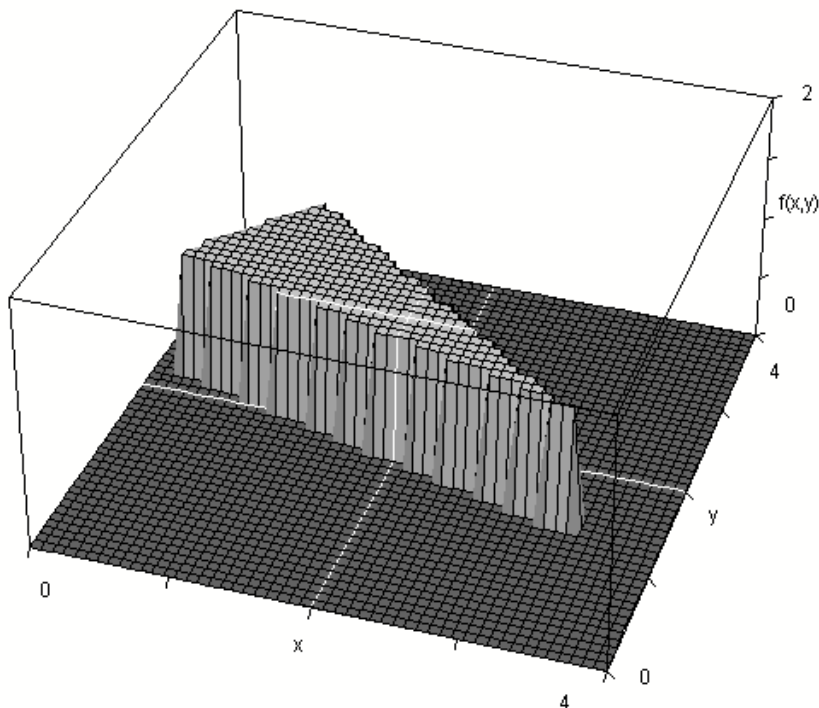


Рисунок 4.2 – Модель просвічуваного шару щільної речовини

За допомогою співвідношень обертання на кут ϕ перетворимо систему координат (x, y) у систему координат $(s, y1)$ з параметром ϕ , для чого

використаємо таке співвідношення

$$[x = s \cdot \cos(\varphi) - y_1 \cdot \sin(\varphi); y = s \cdot \sin(\varphi) + y_1 \cdot \cos(\varphi);$$

$$s = x \cdot \cos(\varphi) + y \cdot \sin(\varphi)].$$

Для повороту функції $f(x, y)$ у систему координат (s, y_1) на кут φ підставимо в неї замість x та y їх вирази через s і y_1 . Конкретні значення нової функції двох змінних $R(s, \varphi, y_1)$ у кожній точці в новій системі координат залежатимуть від параметра φ

$$R(s, \varphi, y_1) := f(s \cdot \cos(\varphi) - y_1 \cdot \sin(\varphi), s \cdot \sin(\varphi) + y_1 \cdot \cos(\varphi)),$$

$$R(s, \varphi, y_1) := \text{SIGN}(\text{SIGN}((210 \cdot s - 170 \cdot y_1) \cdot \cos(\varphi) - (170 \cdot s + 210 \cdot y_1) \cdot \sin(\varphi) + 357) + \text{SIGN}((115 \cdot s + 390 \cdot y_1) \cdot \cos(\varphi) + (390 \cdot s - 115 \cdot y_1) \cdot \sin(\varphi) - 897) - \text{SIGN}((40 \cdot s + 51 \cdot y_1) \cdot \cos(\varphi) + (51 \cdot s - 40 \cdot y_1) \cdot \sin(\varphi) - 204) - 2) / 2 + 1/2 .$$

Формально ця функція є шляхом променя, що проходить крізь тіло просвічуваного шару паралельно координаті y_1 на відстані s від неї, коли нова система координат повернена до вихідної на кут φ .

4.2.2. Формування радонівського образу

У розглянутому прикладі густина речовини по всьому перетину обрана сталою і дорівнює одиниці. Щоб знайти ступінь поглинання енергії променя або ступінь прозорості тіла, необхідно проінтегрувати функцію густини речовини уздовж координати y_1 . Це значення дасть оцінку величини тіні (або яскравості) при заданих значеннях s і φ . У такий спосіб перетворення Радону отримане інтегруванням функції $R(s, \varphi, y_1)$ по координаті y_1 у межах $\pm\infty$, а у випадку розглянутого нами обмеженого перетину шару – у межах ± 5

$$R_s(s, \varphi) := \int (R(s, \varphi, y_1), y_1, -5, 5).$$

Точне вираження для зображення Радону на екрані монітора виводиться досить громіздке. У пам'яті машини воно залишиться до кінця поточної сесії пакета, тому, після побудови тривимірного образу функції $R_s(s, \varphi)$ (рис. 4.3), її вираз з поля алгебри необхідно забрати.

4.2.3. Побудова тривимірних проекцій для кожного кута

Побудова проекцій радонівського образу для заданих кутів просвічування

$$Vrs(s) := \text{VECTOR}(R_s(s, \varphi), \varphi, 0, \pi - \pi/24, \pi/12)$$

Вектор виразів із усіма проекціями з поля Алгебри можна забрати.

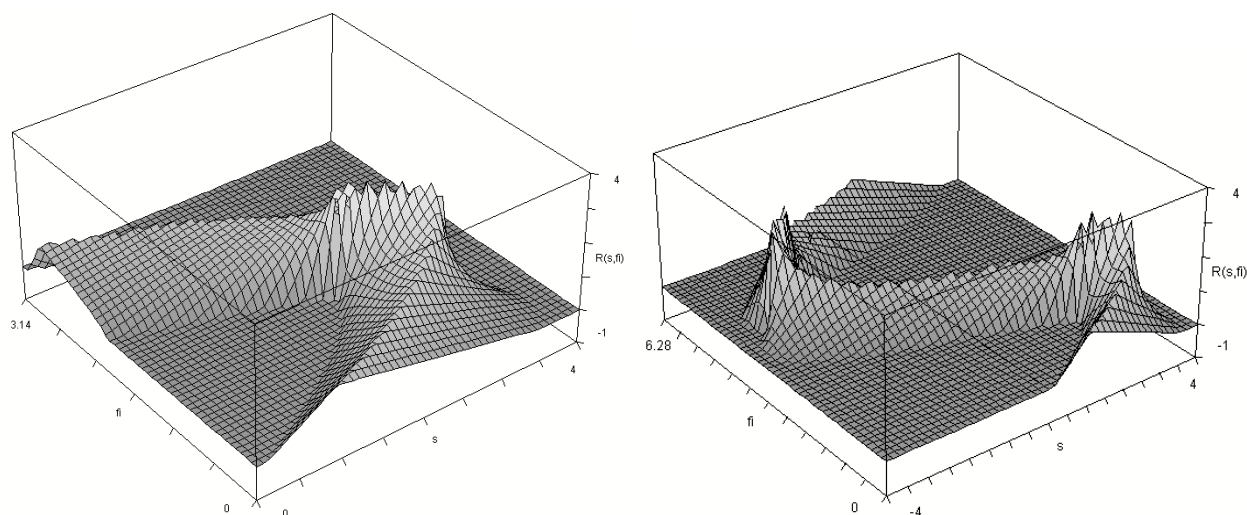


Рисунок 4.3 – Радонівський образ у координатах (s, φ)

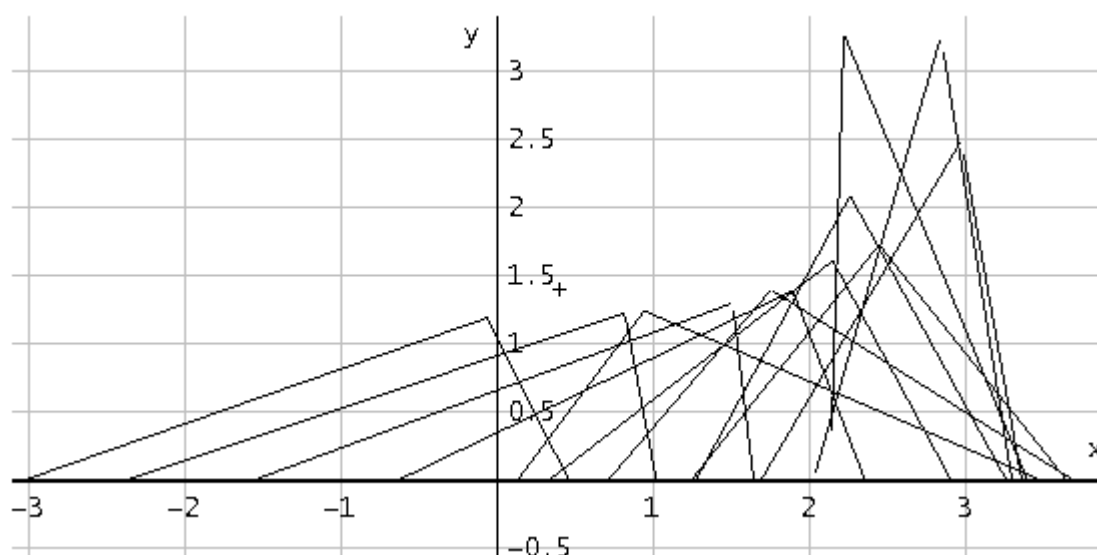


Рисунок 4.4 – Вид 12-и проекцій, видимих через 30° від 0° до 180°

4.2.4. Зворотне перетворення проекцій

До кожної з 12 проекцій знайденого вектора застосовуємо процедуру зворотного проектування, тобто, замість змінної $s(\varphi)$ у кожну із проекцій $Vrs(s)$ підставимо її вираження через x і y , обчислене для відповідного кута φ . Для цього підготуємо 12-и компонентний вектор ss з відповідними значеннями функції $s(\varphi)$

$$ss := \text{VECTOR}(x \cdot \cos(\varphi) + y \cdot \sin(\varphi), \varphi, 0, \pi - \pi/24, \pi/12);$$

$$ss := [x, x \cdot (\sqrt{6}/4 + \sqrt{2}/4) + y \cdot (\sqrt{6}/4 - \sqrt{2}/4), \sqrt{3} \cdot x/2 + y/2, \sqrt{2} \cdot x/2 + \sqrt{2} \cdot y/2, x/2 + \sqrt{3} \cdot y/2, x \cdot (\sqrt{6}/4 - \sqrt{2}/4) + y \cdot (\sqrt{6}/4 + \sqrt{2}/4), y, x \cdot (\sqrt{2}/4 - \sqrt{6}/4) + y \cdot (\sqrt{6}/4 + \sqrt{2}/4),$$

$$\sqrt{3}y/2 - x/2, \sqrt{2}y/2 - \sqrt{2}x/2, y/2 - \sqrt{3}x/2, y*(\sqrt{6}/4 - \sqrt{2}/4) - x*(\sqrt{6}/4 + \sqrt{2}/4)].$$

Тепер в i -ту проекцію необхідно підставити i -е значення вираження ss . Для командного рядка це може мати такий вигляд: $(Vrs(ss \downarrow i)) \downarrow i$. Виконати таку підстановку для всіх проекцій одним оператором можна, якщо в командному рядку сформувати наступне вираження для вектора тривимірних проекцій

$$Vpr:=VECTOR((Vrs(ss \downarrow i)) \downarrow i, i, 1, 12).$$

Після виконання цього оператора відобразимо всі i -і зворотні проекції в тривимірному просторі координатної системи (x,y,z) , що і зображено на рис.4.5:

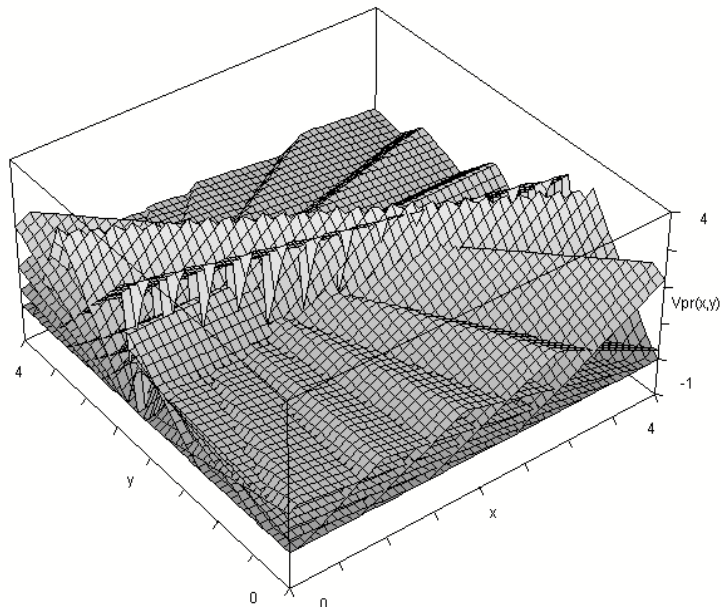


Рисунок 4.5 – Розташування всіх зворотних проекцій

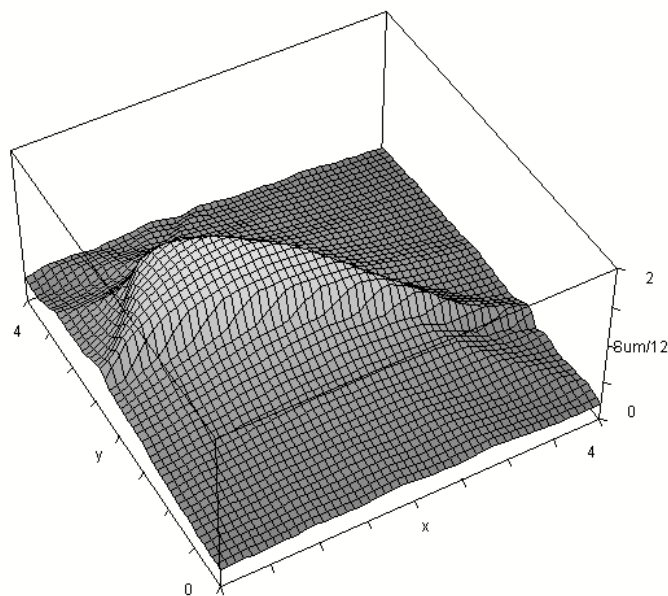


Рисунок 4.6 – Відновлена за радонівському образу густина речовини

4.2.5. Відновлення картини просвіченого шару

Для відновлення внутрішнього розподілу густини речовини досить обчислити середнє арифметичне всіх зворотних радонівських проекцій, помістивши в командний рядок вираження

$$1/12 * \sum(\text{VECTOR}(\text{Vpr} \downarrow k, k, 1, 12)).$$

4.3. Висновки

Проведені дослідження вигляду радонівських образів і способів формування проекцій від просвічування щільного об'єкта під різними кутами дозволили переконатися в працездатності методу зворотного проектування для знаходження розподілу густини речовини в зразку, просвічуваному з усіх боків. Виявлена необхідність, з метою підвищення розподільної здатності, суттєво збільшувати число проекцій, наприклад, на півтора-два порядки порівняно з їхнім числом у здійсненій роботі.

4.4. Порядок виконання роботи

Роботу слід виконувати в пакеті DERIVE v. 6.0 або вище. За заданими параметрами зрізу просвічуваного об'єкта і за пунктами в описаному порядку необхідно зробити математичні перетворення, побудувати графічні зображення, вставляючи їх у текст проведених розрахунків, увести текстові пояснення до кожної назначеної дії та коментарі до одержуваних результатів. У ході виконання роботи звільняти поле Алгебри пакета DERIVE від непотрібних проміжних виразів, залишаючи лише необхідні формули.

До кінця роботи продемонструвати викладачеві на екрані монітора файл лабораторної роботи, зберегати його на носії інформації та на наступному занятті подати її у надрукованому вигляді на аркушах формату А4.

5. Лабораторна робота №3. Обробка зображень і їх гістограм

5.1. Мета роботи

- 1) Згенерувати випадковий двовимірний масив зображення 51×51 і подати його значення у вигляді плоскої райдужної картинки.
- 2) Знайти щільність розподілу та функцію розподілу, як аргументи у яких узято значення яскравості.
- 3) На основі візуального аналізу гістограм навчитись визначати границі областей зайвих чорних і зайвих світлих.
- 4) Навчитись видаляти з гістограм і вихідних зображень малоінформативні тони та перетворювати на весь діапазон значень яскравості частину, що залишилася.
- 5) Освоїти методику нелінійного перетворення яскравості вихідного зображення при різних значеннях гамма-фактору.
- 6) З кожного експерименту проводити візуальне порівняння і на підставі аналізу зробити короткі висновки.

5.2. Виконання роботи

В ході проведення експериментів вибір конкретних параметрів слід узгоджувати з викладачем, тому що вихідні зображення для кожного студента формуються індивідуально. Для настроювання математичного пакета при початковому відкритті встановіть через командний рядок

[Inputmode:=Word, Casemode:=Sensitive, Timesoperator:=Asterisk, Notationdigits:=4].

5.2.1. Підготовка вихідного зображення

У даній роботі як базова функція fs для одержання двовимірного масиву зображення обрана синусоїда, аргумент якої включає три змінні. Значення частоти та фази міняються випадково. Амплітуда, дорівнює

$$(\#e^{(-0.1 \cdot \varphi)}),$$

частота і фаза додатково залежать від параметра φ :

$$fs := \sin(10 \cdot (x \cdot \cos(\varphi) + y \cdot \sin(\varphi)) / (\text{RANDOM}(5) - 2.6) - 1.3 + \text{RANDOM}(7))2.$$

Випадкову функцію rg визначимо сумою чотирьох обчислених виразів

$$(\#e^{(-0.1*\varphi)})*fs,$$

де $\#e$ – основа натурального логарифма; $\varphi \in \{2, 3, 5, 7\}$.

$$rg := \sum(\text{VECTOR}((\#e^{(-0.1*\varphi)})*fs, \varphi, [2, 3, 5, 7])).$$

Функцію rg із двома незалежними змінними x і y перетворимо у двовимірну матрицю розміру 51×51 , задавши змінним y діапазоні $(-2.5 \leq x, y \leq 2.5)$ крок зміни, дорівнює $0,1$. Після цього матрицю перетворимо в матрицю з діапазоном подання цілих чисел від нуля до 255 .

Конкретний текст програми $M(\text{Img})$ з переліченими вище функціями, який копіюється або вводиться в багаторядковий командний рядок, має такий вигляд:

```
M(Img):=
  PROG(
    Img:=[],
    fs:=SIN(5*(x*SIN(φ)+y*COS(φ)))/(0.5*RANDOM(5)-2.6)-
1.3+RANDOM(7))2,
    rg:=Σ(VECTOR(#e(-0.1*φ)*fs,φ,[2,3,5,7])),
    Mz:=VECTOR(VECTOR(rg,x,-2.5,2.5,0.1),y,-2.5,2.5,0.1),
    Img:=FLOOR(255*Mz/MAX(MAX(Mz))),
    RETURN Img)
```

У результаті виконання функції $M(\text{Img})$ буде сформований і виведений на екран безіменний масив цілих чисел. Оскільки кожний виклик функції $M(\text{Img})$ буде формуватиме новий випадковий масив цілих чисел, то рекомендується викликати його кілька разів, надаючи результату нове ім'я, а після їх перегляду залишити картинку, що сподобалася, для виконання лабораторної роботи.

Сказане можна зробити, якщо ввести та викликати на виконання такий вектор, наприклад, на 7 компонент:


$$[A:=M(\text{Img}), B:=M(\text{Img}), C:=M(\text{Img}), D:=M(\text{Img}), E:=M(\text{Img}), \\ F:=M(\text{Img}), G:=M(\text{Img})].$$

Після виконання цього вектора громіздкий результат з екрана Алгебри необхідно відразу ж забрати. Отримані в процесі обчислень випадкові матриці цілих чисел однаково залишаються в пам'яті як значення в іменах матриць: $\{A, \dots, G\}$. Їхні зображення зручно по компонентах переглядати зверху в тривимірній графіці за іменами. Для цього на поле **Алгебри** необхідно вивести

вектор із цими іменами:

[A,B,C,D,E,F,G]

5.2.2. Перегляд плоских картинок у тривимірній графіці

Вибрану курсором кожен компоненту візуально переглядаємо в тривимірній графіці, куди переходимо кнопкою .

У меню **Set⇒Plot⇒Min/max** для x , y і z потрібно встановити відповідно параметри шкал зображуваної матриці: $x=(1,51,10)$; $y=(1,51,10)$; $z=(0,255,51)$. У меню **Set 3D ⇒ Aspect Ratio ⇒ [x,y,z] := [1,1,0.1]**. У меню **Set⇒Plot Parameters⇒Number of Panels:=50**, а в меню **Set⇒Plot Colors :=Rainbow⇒Gray Scale**.

Після завершення установок вивести зображення і маніпулюючи стрілками нахилу і повороту, добитися необхідного плоского зображення і скопіювати його у звіт. Переглянувши всі 7 зображень, вивести у звіт, що сподобався, в плоскому і тривимірному вигляді. Дати нове ім'я **Misx** відібраному зображенню. Для розглянутого прикладу **Misx:=C**.

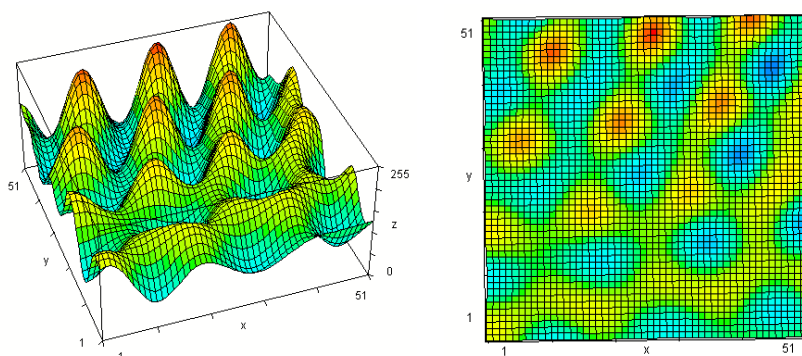


Рисунок 5.1 – Об'ємне та плоске зображення відібраної картини

5.2.3. Побудова гістограми для сірих півтонів

Побудовані матриці 51×51 представляють лише числа градацій сірого. Derive прикрашає їх тільки для зручного читання.

Побудова гістограми щільності розподілу сірих відтінків зображення можна здійснити за допомогою, наприклад, такої програми. У ній готується вектор **H1** з 256 нульовими двохкомпонентними векторами, які будуть представляти координати точок гістограми [абсциса, ордината]. Цикл по k для кожного k переглядає елементи масиву **Misx** (51×51) разів і, якщо $Misx[i,j] = k$, то значення абсциси зберігається, а значення ординати збільшується на

одиницю, інакше абсциса й ордината зберігаються. Текст програми, яка це робить, викликається по імені His(Misx):

```
His(Misx) :=
  PROG(
    q := DIM(Misx),
    H1 := VECTOR(VECTOR(0, j, 1, 2), i, 1, 255),
    k := 1,
    LOOP(
      VECTOR(VECTOR(IF(Misx↓i↓j = k, [H1↓k↓1 := k; H1↓k↓2 := + 1], [H1↓k↓1 := k; H1↓k↓2 := + 0]), i, 1, q), j, 1, q),
      IF(k < 255, k := + 1, exit) ),
    RETURN Gis := H1)
```

Перейшовши у вікно побудови двовимірних графіків, за допомогою установок параметрів вікна і шкал слід досягти для Gis зручного читання зображення гістограми щільності розподілу сірого.

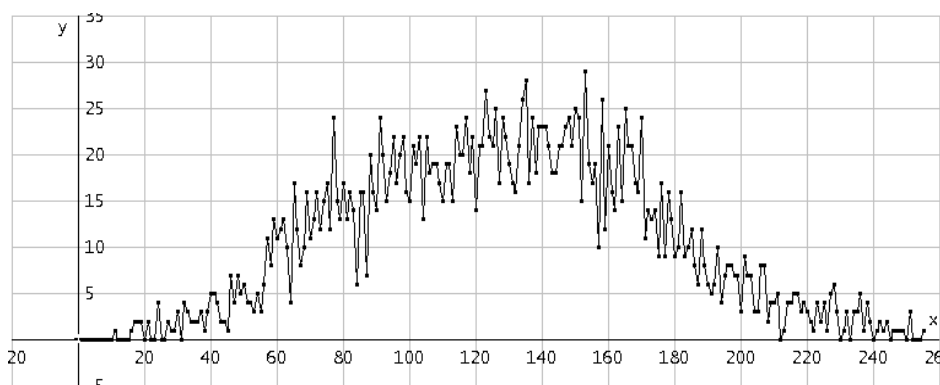


Рисунок 5.2 – Загальний вигляд гістограми

Висвітлену на екран таблицю координат точок після виконання програми з поля Алгебри забрати. В інші оператори вона може передаватися по імені Gis.

5.2.4. Побудова функції розподілу

Аналогічним образом будується програма обчислення функції розподілу сірих відтінків зображення. Відмінність полягає лише в умові перевірки значення сірого в конкретного елемента вихідного зображення Misx.

```
Hcum(Misx) :=
  PROG(
    q := DIM(Misx),
```

```

H2 := VECTOR(VECTOR(0, j, 1, 2), i, 1, 255),
k := 1,
LOOP(
  VECTOR(VECTOR(IF(Misx↓i↓j ≤ k, [H2↓k↓1 := k; H2↓k↓2 :=+ 1], [H2↓k↓1
:= k; H2↓k↓2 :=+ 0]), i, 1, q), j, 1, q),
  IF(k < 255, k :=+ 1, exit) ),
RETURN Cum := H2)

```

Після виклику на виконання програми Hcum(Misx) одержуємо таблицю координат точок з іменем Cum. Для розглянутого зображення Misx функція розподілу має такий вигляд, що надано на рис. 5.3.

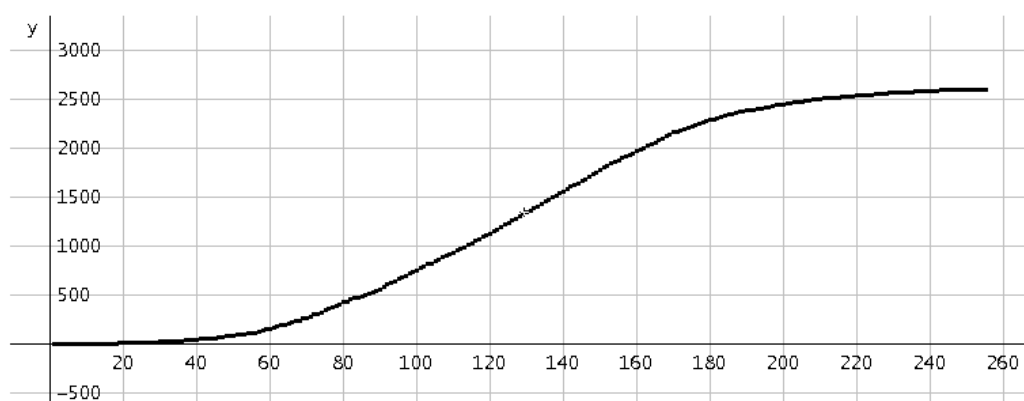


Рисунок 5.3 – Вид інтегральної функції розподілу

Таблицю координат точок після побудови графіка з поля Алгебри забрати.

5.2.5. Перетворення гистограми вихідного зображення

Для усунення малоінформативних ділянок і розтягування значень сірого, що залишилися, на всю шкалу складемо програму, ядром якої будуть дві логічні функції koo і kn, що виконують відповідне вирізання і розтягування на всю шкалу значень сірого. Pozнайомтеся з текстом програми Newm(Misx,kmin,kmax,Mnew), у якій формальні параметри kmin і kmax визначають границі внутрішньої області значень сірого, яку необхідно розтягти на весь діапазон значень (0, 255).

```
Newm(Misx,kmin,kmax,Mnew):=
```

```

PROG(
  Mo:=Misx,
  Mn:=VECTOR(VECTOR(0,i,1,51),j,1,51),
  a_:=255/(kmax-kmin),

```

```

i:=1,
LOOP(
  j:=1,
  LOOP(
    ko:=Mo↓i↓j,
    koo:=IF(ko<kmin,0,IF(kmax<ko,255,a_*(ko-kmin))),
    kn:=FLOOR(IF(koo<0,0,IF(koo>255,255,koo))),
    Mn↓i↓j:=kn,
    IF(j<51, j:+1, exit)),
  IF(i<51, i:+1, exit)),
RETURN Mnew:=Mn,
exit)

```

За завданням викладача для декількох діапазонів яскравості ($kmin$, $kmax$) обріжте краї гістограми вихідного зображення і наведіть об'ємну та плоску картинку результатів цієї обробки, як це показано для діапазону (60, 200).

Указати на видимі відмінності перетвореного зображення від вихідного.

Порівнюючи картинки рисунках 5.1 та 5.4 можна відзначити, наприклад, підвищення контрастності плоского зображення на другому рисунку.

Спробуйте з різних ділянок діапазону значень сірого виконати вирізання таких інтервалів, щоб різниця ($kmax-kmin$) була в межах одиниць, і прокоментуйте результат.

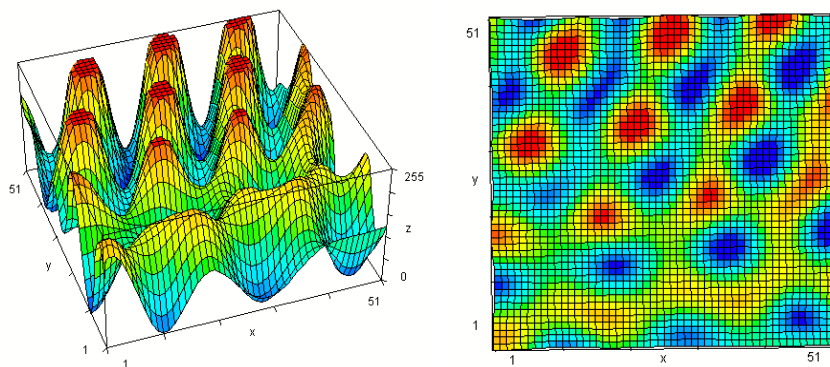


Рисунок 5.4 – Результат перетворення гістограми

5.2.6. Інвертування значень сірого

Інвертування значень сірого часто дозволяє виявити в зображенні додаткові відомості про структурні особливості вихідної картини. Таке перетворення можна виконати такою програмою.

$Minv(Misx, Mnew) :=$

```

PROG(
  Mo := Misx,      Mn := VECTOR(VECTOR(0, i, 1, 51), j, 1, 51),
  i := 1,
  LOOP( j := 1,
    LOOP(
      ko := Mo↓i↓j,      Mn↓i↓j := 255 – ko,
      IF(j < 51, j := j + 1, exit) ),
      IF(i < 51, i := i + 1, exit) ),
  RETURN Mnew := Mn,
  exit)

```

Викликом Minv(Misx, Mnew) одержить інвертований масив і подійте у звіті його зображення і коментарі.

5.3. Нелінійна обробка яскравості

5.3.1. Вид функціональної обробки

Істотну роль в обробці зображень із рівно-щільнісними гістограмами відіграє безперервна зміна сірого за кривими експонентного або логарифмічного вигляду. Частіше функції такого вигляду подають у графічних пакетах виразом

$$knew = \text{FLOOR}(255 \times (kold/255)^\gamma),$$

де зв'язок старого (kold) і нового (knew) значення сірого визначається параметром γ , який може ухвалювати значення, як менше за одиницю (логарифмічний характер кривої), так і більше за одиницю (крива експонентного вигляду). Криві, що спотворюють значення сірого для $\gamma \in \{1/4, 1/3, 1/2, 2, 3, 4\}$, можна вивести для графічної побудови за допомогою оператора (рис. 5.5)

$$\text{VECTOR}(255 * (kold/255)^\gamma, \gamma, [1/4, 1/3, 1/2, 2, 3, 4]).$$

5.3.2. Текст програми гамма-корекції.

Ядром програми служить аналітичний вираз, наведений в попередньому підрозділі.

```

Mgam(Misx, γ, Mnew) :=
  PROG(
    Mo := Misx,
    Mn := VECTOR(VECTOR(0, i, 1, 51), j, 1, 51),

```

```

i := 1,
LOOP( j := 1,
  LOOP( ko := Mo↓i↓j,
    Mn↓i↓j := FLOOR(255*(ko/255)^γ),
    IF(j < 51, j := j + 1, exit) ),
  IF(i < 51, i := i + 1, exit) ),
RETURN Mnew := Mn,
exit)

```

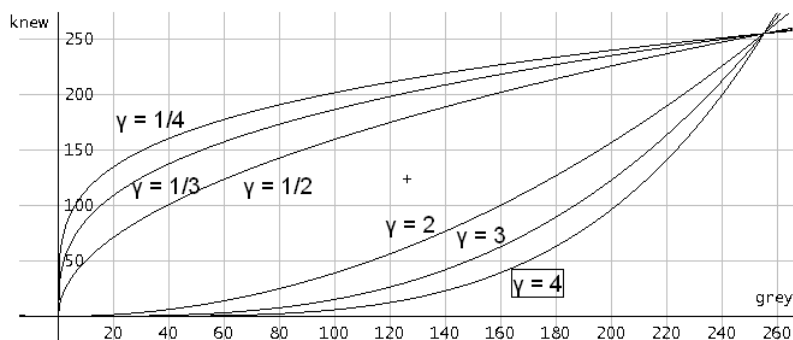
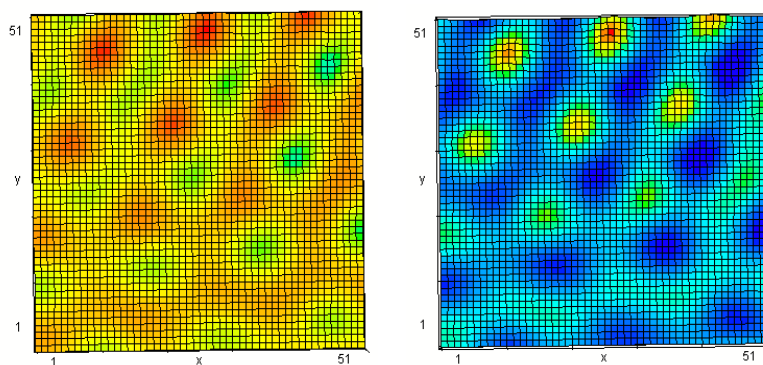


Рисунок 5.5 – Характер залежності значень сірого від γ

У лабораторній роботі виконайте перетворення вихідного зображення для значень параметра гама, зазначеного викладачем. Або, принаймні, для значень гама, дорівняних $1/3$ і 3 . Як приклад для останніх значень γ картинки, відповідно, зображені на рисунку 5.6.



Рисунку 5.6 – Гама корекція для $\gamma=1/3$ (ліворуч) і $\gamma=3$ (праворуч)

5.3.2. Інші види аналітичної корекції гістограм

Криві, що зв'язують нові значення сірого кожного пікселя зображення з їхніми вихідними значеннями, можуть використовувати поліноміальні вираження, складені з різних базисних функцій. Досить цікаві в цьому плані криві Без'є, основою яких є базисні функції Бернштейна. Так, наприклад, базисні функції Бернштейна третьої степені в інтервалі зміни $0 \leq x \leq 1$ мають

такий вигляд

$$[(1-x)^3, 3x(1-x)^2, 3x^2(1-x), x^3] \cdot \text{CHI}(0, x, 1).$$

Функція $\text{CHI}(0, x, 1)$ усередині інтервалу $[0, 1]$ дорівнює 1, а поза інтервалами 0.

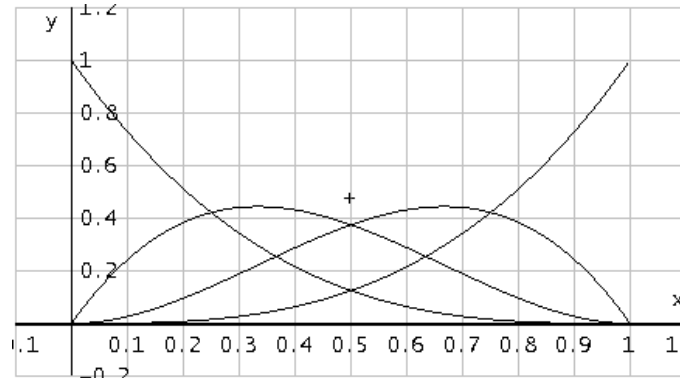


Рисунок 5.7 – Криві Бернштейна третьої степені

Зважена сума цих базисних функцій дозволяє сформувати досить різноманітні криві навіть для вектора вагових коефіцієнтів із двома змінюваними параметрами p_1 і p_2 $[0, p_1, p_2, 1]$.

Вагові коефіцієнти в суму функцій зручно ввести за допомогою скалярного добутку вектора коефіцієнтів і вектора функцій

$$\text{Bz}(p_1, p_2) := [0, p_1, p_2, 1] \cdot [(1-x)^3, 3x(1-x)^2, 3x^2(1-x), x^3] \cdot \text{CHI}(0, x, 1).$$

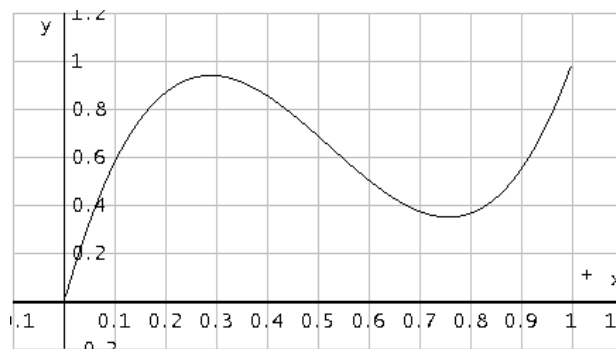


Рисунок 5.8 – Крива Без'є для $(p_1; p_2) = (2, 5; -1)$

У загальному вигляді для двох параметрів $(p_1; p_2)$ крива Без'є має вираження

$$x^2(x^2(3p_1 - 3p_2 + 1) + 3x(p_2 - 2p_1) + 3p_1),$$

яке вставимо в текст програми для перетворення гістограми

$$\text{Mbze}(\text{Misx}, p_1, p_2, \text{Mnew}) :=$$

```

PROG(
  a1:=3*p1-3*p2+1,    a2:=3*(p2-2*p1),    a3:=3*p1,    Mo:=Misx,
  Mn:=VECTOR(VECTOR(0,i,1,51),j,1,51),
  bz(x):=x*(x2*a1+x*a2+a3),    i:=1,    LOOP( j:=1,    LOOP( ko:=Mo↓i↓j,
    Mn↓i↓j:=FLOOR(255*LIM(bz(x),x,ko/255,0)),
    IF(j<51, j:+1, exit)), IF(i<51, i:+1, exit)), RETURN Mnew:=Mn,
  exit).

```

Тепер уведемо цю залежність у програму перетворення значень сірого у вихідному зображенні. Судячи із кривої, значення сірого в діапазоні (0.2, 0.4) будуть збільшені, а в діапазоні (0.7, 0.8) – зменшені. На рисунку 5.9 зображені для порівняння зображення вихідне та перетворене після виконання програми з такими параметрами

Mbze(Misx,2.5,-1,Mnew).

На рис. 5.9 добре помітні почервоніння (посвітління) тих ділянок зображення, які у вихідному зображенні були ближче до синіх (темних).

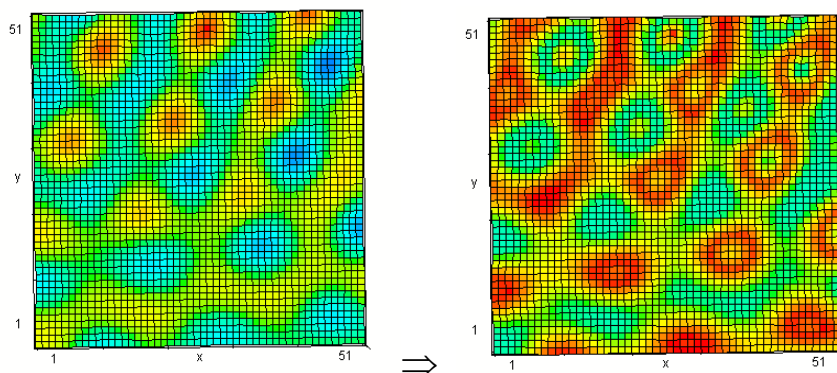


Рисунок 5.9 – Результат корекції кривою Без'є, наведеної на рис. 5.8

5.4. Виконання роботи та оформлення звіту

Роботу слід виконувати в пакеті DERIVE v. 6.0 або вище, сам звіт оформляти в текстовому редакторі Microsoft Word. За заданими параметрами виконати математичні перетворення, побудувати графічні зображення і вставити їх у текст звіту, давати текстові пояснення до кожної намічуваної дії і коментарі до одержуваних результатів. У ході виконання роботи звільняти поле алгебри пакета DERIVE від непотрібних виразів. Не поміщати у звіт текст, де виконуються теоретичні та методичні роз'яснення.

До кінця роботи необхідно продемонструвати викладачеві на екрані монітора файл лабораторної роботи, запам'ятати його на носії інформації та на наступній лабораторній роботі подати її у надрукованому вигляді на аркушах формату A4.

6. Лабораторна робота № 4. Розрахунки та дослідження магнітних полів для стимуляції електромагнітного резонансу атомних ядер

6.1. Мета роботи

Навчитися розраховувати розподіл магнітного поля у заданій області простору. Оцінювати його параметри, важливі для порушення магнітного резонансу в потрібних точках досліджуваного об'єкта, таких як довжина квазілінійної ділянки градієнта поля в заданому зрізі, його дисперсію, діапазон керування напрямком градієнта.

6.2. Порядок виконання роботи

1) Одержати у викладача геометричні характеристики електричних котушок, що задають зріз магнітного поля в об'єкті дослідження відносно ізоцентру магнітної системи та градієнти зміни поля по площині зрізу.

2) Сформувати розрахункові співвідношення для окремих провідників електричних котушок, паралельних координатним осям x і y .

3) Переконавшись у працездатності розрахункових співвідношень, оцінивши візуально одно-, дво- і тривимірні картини поля, спираючись на знання теоретичних основ електротехніки.

4) Сформувати процедури розрахунків поля систем здвоєних котушок (котушок Гельмгольца), що створюють поздовжнє поле зрізу та градієнти поздовжнього магнітного поля по осях x і y . Побудувати тривимірні картини полів кожної системи котушок.

5) Розглянути область лінійності побудованих полів в області ізоцентру магнітної системи. Побудувати графіки вертикальних перерізів у характерних напрямках сумарного поля. Обчислити величини градієнтів та їх дисперсію в межах лінійної зони.

6) Зробити висновок з проведених досліджень.

6.3. Приклад проведення лабораторної роботи

На рисунку 6.1 наведені геометричні розміри плоских котушок: квадратна (а) – для створення поздовжнього (вертикального) поля, що задає горизонтальний зріз у досліджуваному об'єкті; пари прямокутних (б), паралельних своєї довгою стороною до осі x – для створення градієнта поздовжнього поля уздовж осі y і не зображеної на рисунку 6.1 пари аналогічних прямокутних котушок, паралельних осі y – для створення градієнта уздовж осі x . Таких систем з п'ятьма сполученими в одній площині

котушками, симетричними відносно ізоцентру магнітної системи, дві: одна – вище ізоцентру на величину $|z_0|$, інша – на стільки ж нижче.

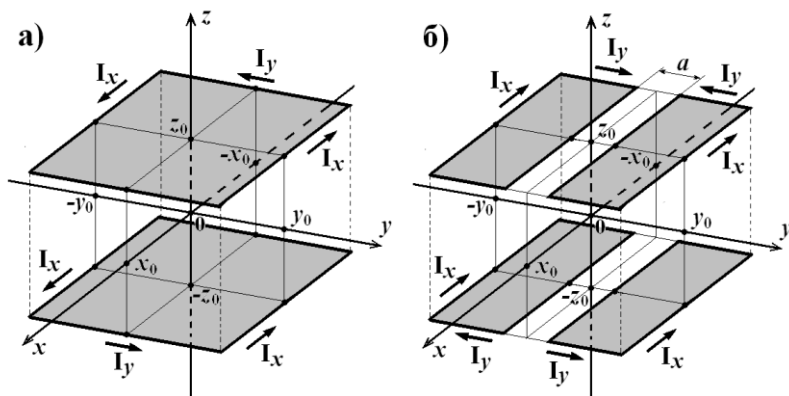


Рисунок 6.1 – Розташування і розміри котушок відносно ізоцентру

Обчислення приросту індукції у довільній точці $\mathbf{M}(x,y,z)$, вилученій від нескінченно малого відрізка $d\mathbf{l}$ провідника зі струмом I на відстань r , проводиться відповідно до закону Біо-Савара-Лапласа за формулою

$$d\mathbf{B} = \frac{\mu \mu_0}{4\pi} \cdot \frac{I \cdot (d\mathbf{l} \times \mathbf{r})}{r^3} = \frac{\mu \mu_0}{4\pi} \cdot \frac{I}{r^2} \cdot |d\mathbf{l}| \cdot \sin(\angle(d\mathbf{l}, \mathbf{r})), \quad \left[\mathbf{B} = \frac{\mathbf{B} \cdot \mathbf{c}}{\text{м}^2} \right]$$

де $d\mathbf{B}$, $d\mathbf{l}$, \mathbf{r} – векторні величини; r – довжина вектора \mathbf{r} у метрах;

I – амплітуда струму в амперах; $\angle(d\mathbf{l}, \mathbf{r}) = \alpha$ – кут між векторами.

Для обчислення сумарної індукції у заданій точці від усіх елементів проведення зі струмом зручно пов'язати $|d\mathbf{l}|$, α і $|r| = r$ з координатою точки \mathbf{C} у \mathbf{C} . На рисунку 6.2 наведені геометричні побудови, що описують походження параметрів і величин, використовуваних нижче у формулах розрахунків магнітної індукції в точці $\mathbf{M}(x,y,z)$, яка викликається струмом I_y , що протікають паралельно осі y по провідникові від точки \mathbf{A} до точки \mathbf{B} .

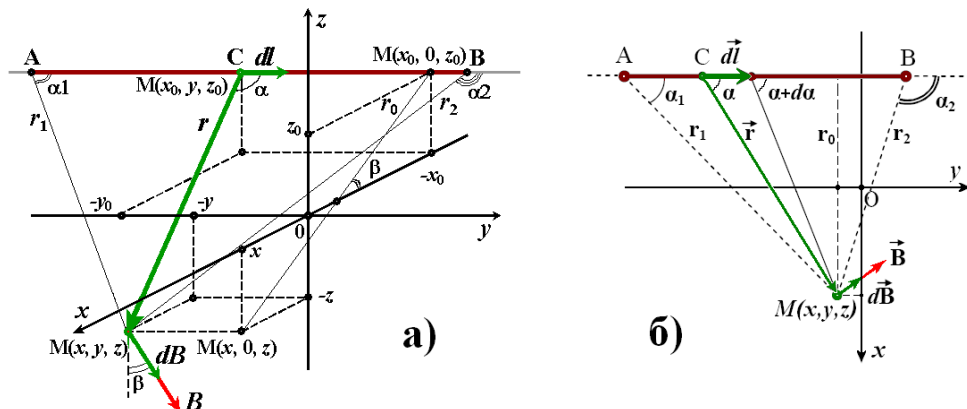


Рисунок 6.2 – Геометричні параметри провідника зі струмом у точці $\mathbf{M}(x,y,z)$

6.4. Вибір варіанта роботи

Як базовий розмір системи електричних котушок оберемо параметр a , який в остаточному підсумку має визначати об'єм області для розміщення досліджуваного об'єкта. Величина параметра в метрах обчислюється так

$$a = 0.02 + 0.001 \cdot (N \bmod(10)) \text{ м},$$

де N – порядковий номер студента в журналі академічної групи.

Інші параметри котушок обчислюються у такий спосіб.

1) Для керування магнітним полем по вертикалі (вибір перетину)

$$|x_0| = |y_0| = 3.5a, |z_0| = 0.75a.$$

2) Для керування величиною градієнта магнітного поля по осі x

$$|x_0| = |y_0| = 2a, |z_0| = 0.75a.$$

3) Для керування величиною градієнта магнітного поля по осі y

$$|x_0| = |y_0| = 2.25a, |z_0| = 0.75a.$$

6.4.1. Визначення обчислювальних операторів у середовищі Derive 6.1

При відкритті пакета для початкових установок уведіть вектор

[Inputmode:=Word, Casemode:=Sensitive,
Timesoperator:=Asterisk, Notationdigits:=4].

Для прикладу значення базового параметра a обчислимо для варіанта $N=0$

$$a = 0.02 + 0.001 \cdot (0 \cdot \bmod(10)) = 0.02 \text{ м}.$$

Для провідника зі струмом I_y , який починається в точці A і закінчується в точці B (рис. 6.2 а), будемо використовувати наступні позначення:

yn – координата точки A відносно ізоцентру системи котушок;

yk – координата точки B відносно ізоцентру системи котушок;

(x_0, z_0) – точка перетинання прямої лінії із площиною xOz .

Найкоротша відстань від провідника зі струмом до точки $M(x, y, z)$

$$r_0 = \sqrt{(x_0 - x)^2 + (z_0 - z)^2}; \quad "r0y:=\sqrt{((x0-x)^2+(z0-z)^2)}."$$

У подвійних лапках за формулами будемо поміщати вирази, які без лапок можна безпосередньо переносити в командний рядок пакета Derive 6.1.

Відстань від точки **C** на прямій зі струмом до точки **M(x, y, z)**:

$$r = \sqrt{(y - y_C)^2 + r_0^2}; \quad \text{"ry:=}\sqrt{((y-y_C)^2+r_0^2)}\text{"}$$

Синус кута α між провідником (рис. 6.2 б) і вектором **r** дорівнює відношенню модуля його проекції на площину **xOz** до модуля **r**, або:

$$\alpha = \arcsin\left(\frac{|r_0|}{|r|}\right); \quad \text{"}\alpha\text{y:=ASIN(ABS(r0y)/ABS(ry))}\text{"}$$

Для провідника паралельного осі **y**, $dl = dy_C$, а синус кута α , який входить у вираз для приросту індукції в точці **M(x, y, z)**, можна уявити так:

$$\sin\left(\arcsin\left(\frac{|r_0|}{|r|}\right)\right); \quad \text{"SIN(ASIN(ABS(r0y)/ABS(ry)))"}.$$

У результаті, підінтегральний вираз, що є функцією y_C , ухвалює такий вигляд

$$\frac{\sin\left(\arcsin\left(\frac{|r_0|}{|r|}\right)\right)}{\left(\sqrt{(y - y_C)^2 + r_0^2}\right)^2}; \quad \text{"SIN(ASIN(ABS(r0y)/ABS(ry)))/(\sqrt{((x0-x)^2+(z0-z)^2))^2"}.$$

Тепер отриману залежність від y_C можна проінтегрувати по y_C в межах зміни y_C , тобто, від **yn** до **yk**. Помноживши цей інтеграл на масштабуючий множник $\frac{\mu \mu_0}{4\pi} \approx 10^{-7}$ і величину струму **Iy**, що протікає по провідникову, одержимо значення вектора індукції в точці **M(x, y, z)**. Цей вектор перпендикулярний площині, у якій лежить провідник зі струмом і точка **M(x, y, z)**, і спрямований так, що з його кінця поворот вектора елемента довжини **dl** убік сполучення останнього з радіусом-вектором **r** відбувається проти годинникової стрілки.

$$\mathbf{B} = \frac{\mu \mu_0}{4\pi} \cdot I_y \cdot \int_{y_n}^{y_k} \frac{\sin\left(\arcsin\left(\frac{|r_0|}{|r|}\right)\right)}{\left(\sqrt{(y - y_C)^2 + r_0^2}\right)^2} dy_C ;$$

$$"10^{(-7)} \cdot I_y \cdot \int (\sin(\arcsin(\sqrt{((x_0-x)^2 + (z_0-z)^2)} / \sqrt{((y-y_C)^2 + r_0^2)}))) / \sqrt{((x_0-x)^2 + (z_0-z)^2)}^2, y_C, y_n, y_k)"$$

Кут нахилу площини, до якої перпендикулярний вектор \mathbf{B} , а значить, і кут між ним і вертикальною віссю z рівняє

$$\beta = \arccos\left(\frac{x_0 - x}{\sqrt{(x_0 - x)^2 + (z_0 - z)^2}}\right); \quad \beta_y := \arccos((x_0 - x) / \sqrt{((x_0 - x)^2 + (z_0 - z)^2)}).$$

Щоб одержати вертикальну проекцію B_z вектора \mathbf{B} , необхідно значення останнього помножити на $\cos(\beta)$.

$$"10^{(-7)} \cdot I_y \cdot \int (\sin(\arcsin(\sqrt{((x_0-x)^2 + (z_0-z)^2)} / \sqrt{((y-y_C)^2 + r_0^2)}))) / \sqrt{((x_0-x)^2 + (z_0-z)^2)}^2, y_C, y_n, y_k) \cdot \cos(\beta_y)"$$

Поданий вираз визначимо як тривимірну функцію вертикальної складової індукції $B_{yz}(\dots)$ у точці $\mathbf{M}(x, y, z)$ з параметрами, що визначають довжину провідника (y_n, y_k), його розташування в тривимірному просторі (x_0, z_0) і струм I_y у провіднику.

$$"B_{yz}(x, y, z, y_n, y_k, x_0, z_0, I_y) := 10^{(-7)} \cdot I_y \cdot \int (\sin(\arcsin(\sqrt{((x_0-x)^2 + (z_0-z)^2)} / \sqrt{((y-y_C)^2 + r_0^2)}))) / \sqrt{((x_0-x)^2 + (z_0-z)^2)}^2, y_C, y_n, y_k) \cdot \cos(\beta_y)"$$

Виклик функції здійснюється по її імені зі списком конкретних значень формальних параметрів: $B_{yz}(x, y, z, y_n, y_k, x_0, z_0, I_y)$.

6.4.2. Визначення функції для індукції, паралельної осі x

Модифіковані записи виразів для командного рядка під провідник, паралельний осі x , одержують такий вигляд:

$$"r_{0x} := \sqrt{((y_0 - y)^2 + (z_0 - z)^2)} \quad "r_x := \sqrt{((x - x_C)^2 + r_{0x}^2)}.$$

$$"\alpha_x := \arcsin(\sqrt{r_{0x}} / \sqrt{r_x})". \quad " \sin(\arcsin(\sqrt{r_{0x}} / \sqrt{r_x}))".$$

$$"\sin(\arcsin(\sqrt{r_{0x}} / \sqrt{r_x})) / (\sqrt{((y_0 - y)^2 + (z_0 - z)^2)})^2".$$

$$"10^{(-7)} \cdot I_x \cdot \int (\sin(\arcsin(\sqrt{((y_0 - y)^2 + (z_0 - z)^2)} / A$$

$$BS(\sqrt{((x-xc)^2+r0x^2)}))/\sqrt{((y0-y)^2+(z0-z)^2)^2,xс,xn,xk})"$$

$$"\beta x:=ACOS((y0-y)/\sqrt{((y0-y)^2+(z0-z)^2)})."$$

$$"10^{(-7)}*lx*\int(SIN(ASIN(ABS(\sqrt{((y0-y)^2+(z0-z)^2)}))/ABS(\sqrt{((x-xc)^2+r0x^2)}))/\sqrt{((y0-y)^2+(z0-z)^2)^2,xс,xn,xk)*COS(\beta x)".$$

$$"Bxz(x,y,z,xn,xk,y0,z0,lx):=10^{(-7)}*lx*\int(SIN(ASIN(ABS(\sqrt{((y0-y)^2+(z0-z)^2)}))/ABS(\sqrt{((x-xc)^2+r0x^2)}))/\sqrt{((y0-y)^2+(z0-z)^2)^2,xс,xn,xk)*COS(\beta x)".$$

$$"Bxz(x,y,z,xn,xk,y0,z0,lx)".$$

6.4.3. Розрахунки розподілу вертикальної складової поля для співвісної системи квадратних котушок

Знак струму завбільшки 1 А, який протікає по провіднику, вибираємо позитивним, якщо його напрямок збігається з напрямком відповідної координатної осі, а якщо ні негативним.

$$"W1(x,y,z):=[Bxz(x,y,z,-0.035,0.035,-0.035,0.015,-1); Bxz(x,y,z,-0.035,0.035,0.035,0.015,+1);Byz(x,y,z,-0.035,0.035,-0.035,0.015,-1); Byz(x,y,z,-0.035,0.035,0.035,0.015,+1)]"$$

$$"W2(x,y,z):=[Bxz(x,y,z,-0.035,0.035,-0.035,-0.015,-1);Bxz(x,y,z,-0.035,0.035,0.035,-0.015,+1);Byz(x,y,z,-0.035,0.035,-0.035,-0.015,-1); Byz(x,y,z,-0.035,0.035,0.035,-0.015,+1)]"$$

Для зручності спостереження малих значень поля зрізу (порядку 10^{-5}) уведемо у вираження поля для квадратних котушок множник, що дорівнює 10^5 :

$$"Zbz(x,y,z):=10^5*(\sum((W1(x,y,z))\downarrow i\downarrow 1,i,1,4)+\sum((W2(x,y,z))\downarrow i\downarrow 1,i,1,4))".$$

Побудуємо картину горизонтального розподілу поля в ізоцентрі магнітної системи $Zbz(x,y,0)$ (рис. 6.3).

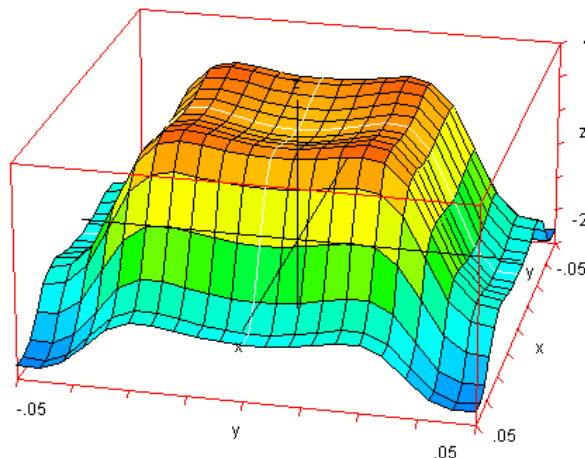


Рисунок 6.3 – Картина поля горизонтального зрізу

Для дослідження рівномірності розподілу поля в робочій області, що має об'єм приблизно $(a \times a \times a)$ м³, зробимо перетин цієї поверхні вертикальною площиною yOz для двох значень $z=\{0, a/2\}$.

$$[Zbz(x,0,0); Zbz(x,0,0.01)]$$

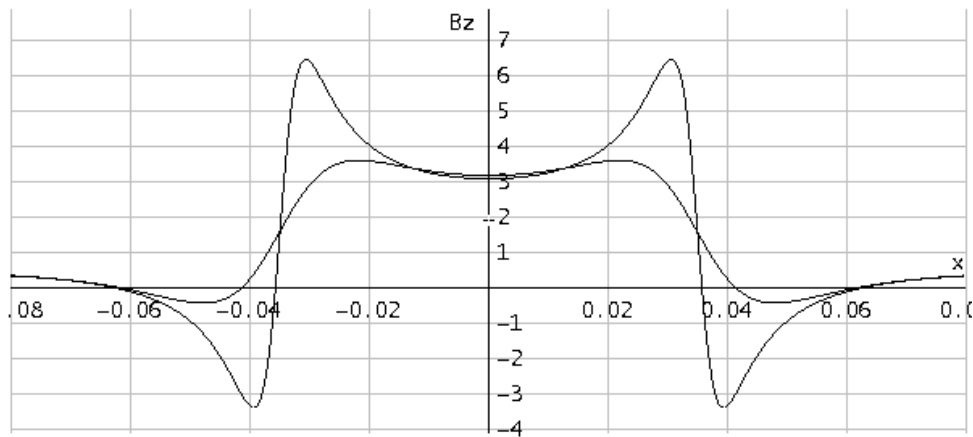


Рисунок 6.4 – Розподіл горизонтального поля уздовж осі x на центральному ($z=0$) і крайньому ($z=0,01$) рівнях робочої області

Із графіків видно, що для робочої області $a/2 = \pm 0.01$ криві майже накладаються один на одну. Щоб оцінити неоднорідність поля в робочій області, виконаємо методом найменших квадратів апроксимацію ділянки кривих за 11 точками, рівномірно розташованими в робочій області:

$$\begin{aligned} & \text{"[M1:=VECTOR([y,Zbz(0,y,0)],} \\ & y,-0.0125,0.0125,0.0025); M2:=VECTOR([y,Zbz(0,y,0.01)],} \\ & y,-0.0125,0.0125,0.0025); Cu1:=FIT([v,a0],M1);} \\ & Cu2:=FIT([v,a0],M2); \sigma1:=\sqrt{(\sum((M1 \downarrow i \downarrow 2 - cu1)^2, i, 1, 11)/11)}; \\ & \sigma2:=\sqrt{(\sum((M2 \downarrow i \downarrow 2 - cu2)^2, i, 1, 11)/11)} \text{"}. \end{aligned}$$

У цьому наборі операторів $M1$, $M2$ – матриці з 11-ю точками; $Cu1$, $Cu2$ – поліноми нульової степені; $\sigma1$, $\sigma2$ – середньоквадратичні відхилення реальних кривих від прямих. Для даного варіанта:

$$\begin{aligned} Cu1 &:= 3.258644115; & Cu2 &:= 3.194678245; \\ \sigma1 &:= 0.06982330139; & \sigma2 &:= 0.1021959811; \\ [\sigma1/Cu1, \sigma2/Cu2] &:= [0.02142710247, 0.03198944409]. \end{aligned}$$

6.4.4. Розрахунки розподілу поля для системи градієнтних котушок

Спочатку складемо описи полів, створюваних окремими котушками системи, що формують градієнт вертикальної складової уздовж осі y :

QLH(x,y,z) – котушка ліва верхня (рис. 6.1 б);

QLL(x,y,z) – котушка ліва нижня;

QRH(x,y,z), QRL(x,y,z) – праві котушки верхні і нижні.

Обчислення вертикальних складових поля для будь-якого провідника кожної котушки здійснюється функціями $Bxz(\dots)$ і $Byz(\dots)$:

```
"QLH(x,y,z):=[Bxz(x,y,z,-0.04,0.04,-0.04,0.015,+1);
Byz(x,y,z,-0.04,-0.01,0.04,0.015,+1);Bxz(x,y,z,-0.04,0.04,-0.01,0.015,-1);
Byz(x,y,z,-0.04,-0.01,-0.04,0.015,-1)]"
"QLL(x,y,z):=[Bxz(x,y,z,-0.04,0.04,-0.04,-0.015,+1);
Byz(x,y,z,-0.04,-0.01,0.04,-0.015,+1);
Bxz(x,y,z,-0.04,0.04,-0.01,-0.015,-1);Byz(x,y,z,-0.04,-0.01,-0.04,-0.015,-1)]"
"QRH(x,y,z):=[Bxz(x,y,z,-0.04,0.04,0.01,0.015,-1);
Byz(x,y,z,0.01,0.04,0.04,0.015,-1);
Bxz(x,y,z,-0.04,0.04,0.04,0.015,+1);Byz(x,y,z,0.01,0.04,-0.04,0.015,+1)]"
"QRL(x,y,z):=[Bxz(x,y,z,-0.04,0.04,0.01,-0.015,-1);
Byz(x,y,z,0.01,0.04,0.04,-0.015,-1);
Bxz(x,y,z,-0.04,0.04,0.04,-0.015,+1);Byz(x,y,z,0.01,0.04,-0.04,-0.015,+1)]"
```

Сумарне магнітне поле із градієнтом уздовж осі y одержуємо так:

```
"Ybz(x,y,z):=10^5*(Σ((QLH(x,y,z))↓i↓1,i,1,4)+Σ((QLL(x,y,z))↓i↓1,i,1,4))+10^5*(
Σ((QRH(x,y,z))↓i↓1,i,1,4)+Σ((QRL(x,y,z))↓i↓1,i,1,4))"
"Ybz(x,y,0)"
```

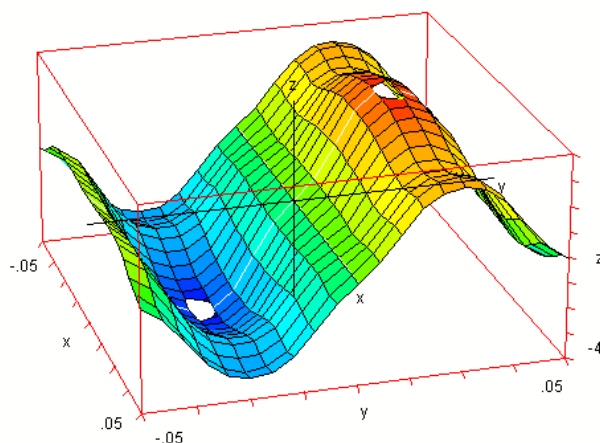


Рисунок 6.5 – Картина поля із градієнтом уздовж осі y

Візьмемо вертикальний переріз градієнтної картини поля уздовж осі y

$$"Sy(v):=Ybz(0,v,0)".$$

Сформуємо масив з 11 координат точок отриманого перетину

$$"Q1:=VECTOR([v,Ybz(0,v,0)],v,-0.0125,0.0125,0.0025)".$$

Тепер за Q1 знаходимо апроксимуючу криву першої степені:

$$"Cy1(v):=FIT([v,a0+a1*v],Q1)"; \quad Cy1(v)=161.7270614 \cdot v.$$

На рисунку 6.6 зображені $Sy(v)$, 11 точок Q1 і пряма $Cy1(v)$.

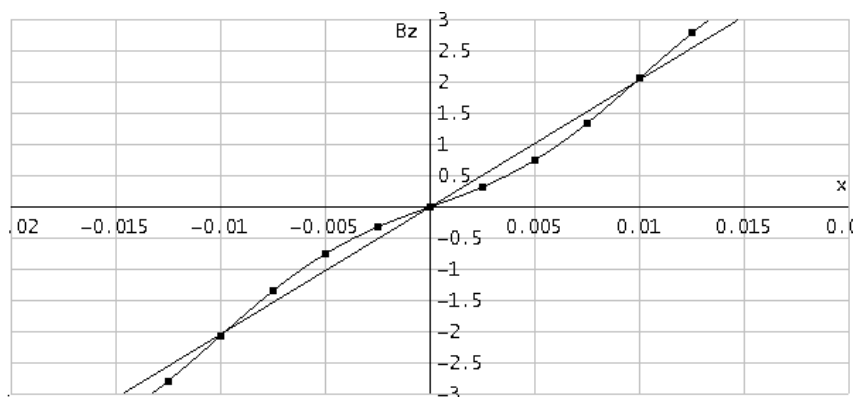


Рисунок 6.6 – Градієнтні криві уздовж осі y в області ізоцентру

Середньоквадратичне відхилення кривої перетину поля від прямої лінії обчислимо по 11 точках різниці цих ліній:

$$"Vdif:=VECTOR([v,Sy(v)-Cy1(v)],v,-0.0125,0.0125,0.0025)";$$

$$"sy:=\sqrt{(1/11 \cdot \sum(Vdif_i^2))}"; \quad sy := 0.1899382385.$$

Розраховані значення поля і його відхилень від ізоцентру для зручності були збільшені в 10^5 разів і віднесені до величини струму в 1А. Градієнт поля дорівнює тангенсу кута нахилу апроксимуючої прямої. Для системи котушок після зменшення в 10^5 разів градієнт складатиме 2.038757546 [мВб/(А·м)]. Збільшенням струму і/або числа витків у котушці можна одержати необхідну величину градієнта поля.

6.4.5. Розрахунки для котушок, що створюють градієнт уздовж осі x

Для котушок, орієнтованих на вісь осі y довгими сторонами, опис поля сформуємо з описів поля таких котушок:

QNH(x,y,z) – котушка негативної півплощини верхня (рис. 6.1, б);

QNL(x,y,z) – котушка негативної півплощини нижня;

QRH(x,y,z), QPL(x,y,z) – котушки позитивної півплощини.

Обчислення вертикальних складових поля для будь-якого провідника кожної котушки, здійснюється тими ж функціями $Vxz(...)$ і $Vyz(...)$:

```
"QNH(x,y,z):=[Byz(x,y,z,-0.045,0.045,-0.01,0.015,-1);Bxz(x,y,z,-0.045,-0.01,0.045,0.015,+1);Byz(x,y,z,-0.045,0.045,-0.045,0.015,+1);Bxz(x,y,z,-0.045,-0.01,-0.045,0.015,-1)]"
```

```
"QNL(x,y,z):=[Byz(x,y,z,-0.045,0.045,-0.01,-0.015,-1);Bxz(x,y,z,-0.045,-0.01,0.045,-0.015,+1);Byz(x,y,z,-0.045,0.045,-0.045,-0.015,+1);Bxz(x,y,z,-0.045,-0.01,-0.045,-0.015,-1)]"
```

```
"QPH(x,y,z):=[Byz(x,y,z,-0.045,0.045,0.045,0.015,+1);Bxz(x,y,z,0.01,0.045,0.045,0.015,-1);Byz(x,y,z,-0.045,0.045,0.01,0.015,-1);Bxz(x,y,z,0.01,0.045,-0.045,0.015,+1)]"
```

```
"QPL(x,y,z):=[Byz(x,y,z,-0.045,0.045,0.045,-0.015,+1);Bxz(x,y,z,0.01,0.045,0.045,-0.015,-1);Byz(x,y,z,-0.045,0.045,0.01,-0.015,-1);Bxz(x,y,z,0.01,0.045,-0.045,-0.015,+1)]"
```

Сумарне магнітне поле із градієнтом уздовж осі x одержуємо так:

```
"Xbz(x,y,z):=10^5*(Σ((QPH(x,y,z))↓i↓1,i,1,4)+Σ((QPL(x,y,z))↓i↓1,i,1,4))+10^5*(Σ((QNH(x,y,z))↓i↓1,i,1,4)+Σ((QNL(x,y,z))↓i↓1,i,1,4))"
```

Картина поля, яка має відносно ізоцентру градієнт вертикальної складової уздовж осі x , виходить шляхом виклику функції "Xbz(x,y,0)" з параметром $z=0$ (рис. 6.7).

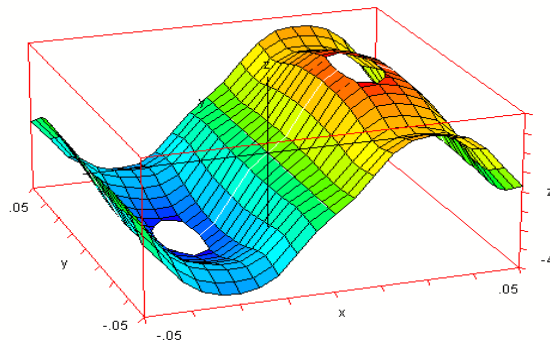


Рисунок 6.7 – Градієнт поля уздовж осі x

Тепер побудуємо картину поля з таким градієнтом по осях x і y , щоб результуючий градієнт не змінив величини і був спрямований під кутом φ . Загальна картина розподілу поля в просторі навколо ізоцентру може бути представлена таким виразом командного рядка

```
"Xybz(x,y,z,φ):=Xbz(x,y,z)*SIN(φ)+Ybz(x,y,z)*COS(φ)".
```

На рис. 6.8 подається картина поля в області ізоцентру з градієнтом, спрямованим під кутом 45° у площині xOy .

Зміну величини градієнта поля при різних кутах φ одержимо за допомогою функції $S\varphi(v, \varphi)$ для декількох значень $\pi/20 \leq \varphi \leq \pi/4$

```
"Sφ(v,φ):=Xybz(v,v,0,φ)" "VECTOR([Sφ(v,φ)],φ,pi/20,pi/4,pi/20)".
```

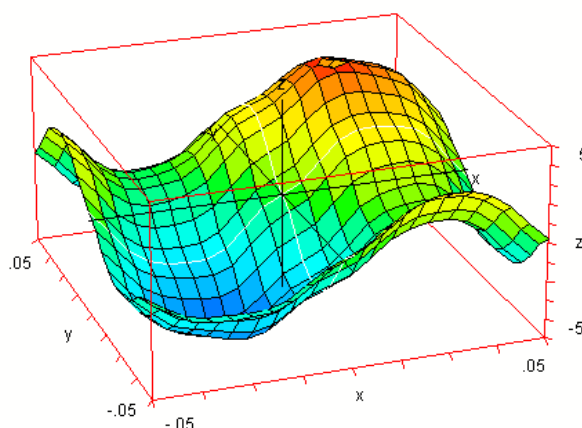


Рисунок 6.8 – Градієнт спрямовано під кутом 45°

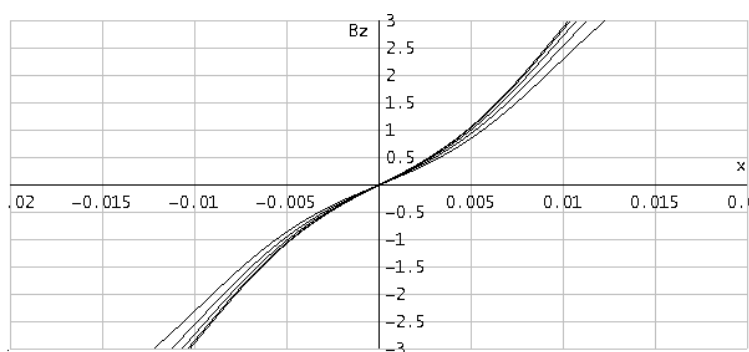


Рисунок 6.9 – Криві градієнта при кутах від 9° до 45°

6.5. Результати дослідження

Виконана лабораторна робота підтвердила можливість одержання градієнтів вертикальної складової магнітного поля будь-якого напрямку за допомогою прямокутних електричних котушок, що оточують задану робочу область. У даному варіанті геометричних розмірів на одиницю струму в 1 ампер отриманий лінійний градієнт поля, приблизно дорівнює $G=2.04$ [мВб/(А·м)] із середньоквадратичним відхиленням порядку $\sigma \approx 0.2$ [мВб/(А·м)].

Також в ході роботи набути навички з розрахунків електричних магнітних систем і їх глибокого дослідження магнітних полів за допомогою професійного пакета символічної математики DERIVE 6.1.

6.6. Оформлення звіту

У звіті навести побудови базових функцій в операторах пакета. Сформулювати для свого варіанта числові параметри для функцій магнітного поля кожного провідника. Зобразити всі картини полів і графіків, що характеризують проведені розрахунки і дослідження.

7. Лабораторна робота №5. Моделювання рівнянь Блоха

7.1. Мета роботи

Досліджувати динаміку вектора макроскопічної намагніченості речовини, поміщеного в зовнішнє магнітне поле, шляхом стимулювання в ньому процесів ядерного магнітного резонансу. Оцінити вплив параметрів речовини, зовнішнього поля й радіочастотного опромінення на характер еволюції вектора намагніченості. Навчитися будувати обчислювальні моделі динамічних процесів у пакеті моделювання аналого-цифрових систем Мікро-Сар-9: створювати обчислювальні схеми, задавати параметри її компонентам, задавати параметри обчислювальному процесу, управляти графічно результатами.

7.2. Теоретичні відомості

Математична модель поведінки ядерної макроскопічної намагніченості в магнітних полях була вперше побудована Ф. Блохом в 1946 році. У даній роботі для обертової із частотою Лармора системи координат, як математичну модель еволюції вектора намагніченості будемо використовувати систему диференціальних рівнянь такого вигляду

$$\begin{cases} \frac{dM_x}{dt} + \frac{M_x}{T_2} + \Delta\omega_z M_y = 0, \\ \frac{dM_y}{dt} + \frac{M_y}{T_2} - \Delta\omega_z M_x + (\gamma_p \tilde{B}_x) M_z = 0, \\ \frac{dM_z}{dt} + \frac{M_z}{T_1} - \frac{M_0}{T_1} - (\gamma_p \tilde{B}_x) M_y = 0, \end{cases} \quad (7.1)$$

де $\mathbf{M}_V = (M_x, M_y, M_z)$ – вектор макроскопічної намагніченості, поданий трьома проекціями в прямокутній обертовій системі координат, з яких M_z спрямована уздовж зовнішнього магнітного поля; \mathbf{M}_0 – вектор початкової макроскопічної намагніченості речовини; T_1 , T_2 – параметри спин-граткової і спин-спінової спонтанної взаємодії атомів з енергією навколишньої речовини, які називають постійними часу поздовжньої і поперечної релаксації; $(\gamma_p \tilde{B}_x) = \omega$ – поляризоване уздовж осі x радіочастотне поле з амплітудою \tilde{B}_x і круговою частотою ω , дорівнює частоті Лармора ω_0 ; $\Delta\omega_z = \omega - \omega_0$ – величина розстроювання частоти накачування відносно частоти ядерного магнітного резонансу.

Перш ніж будувати обчислювальну модель, необхідно вихідну математичну модель представити у відносних безрозмірних змінних. Це дозволить працювати із числовими значеннями шуканих змінних у діапазоні одиниць. Як безрозмірні змінні використовуємо

$$M_0 = \frac{M_0}{M_0} = 1, M_x = \frac{M_x}{M_0}, M_y = \frac{M_y}{M_0}, \quad M_z = \frac{M_z}{M_0}.$$

Масштаб для незалежної змінної часу задамо відношенням

$$m_t = \frac{\Delta t_p}{\Delta t_m} = \frac{\omega_m}{\omega_p},$$

де t_m, ω_m – представлення часу й частоти в обчислювальній моделі;

t_p, ω_p – значення часу й кутової частоти в математичній моделі.

Для обчислення масштабу часу візьмемо як одиницю виміру тривалості реального процесу $\Delta t_p = T_1$ секунд і надамо їй відповідну тривалість в обчислювальній моделі $\Delta t_m = 1$ с. Значення масштабу часу дорівнюватиме $m_t = T_1$.

Виразивши t_p, ω_p через t_m, ω_m, m_t і підставивши їх у вихідну систему (7.1), одержимо систему рівнянь у змінних обчислювальної моделі такого вигляду:

$$\begin{cases} M_x' = -\frac{T_1}{T_2} M_x - \omega_m k M_y, \\ M_y' = -\frac{T_1}{T_2} M_y + \omega_m k M_x - \omega_m M_z, \\ M_z' = -M_z + M_0 + \omega_m M_y, \end{cases} \quad (7.2)$$

де $k = \frac{\Delta \omega}{\omega_0}$ – коефіцієнт відносного розстроювання частоти, зміни якого

зручно вибрати в такому діапазоні $-0.01 < k < 0.01$.

Для аналізу поведінки вектора макроскопічної намагніченості, викликаного переорієнтацією спінів протонів в активних атомах водню, задамо фізичні константи для досліджуваної органічної речовини:

- $\gamma_p = 42,576375 \times 10^6$ [Гц/Тл] – гіромагнітне відношення протонів

атомах водню;

- сталі часу поздовжньої і поперечної складових, що дорівнюють $T_1 = 0,5 \cdot 10^{-3}$ с і $T_2 = 0,25 \cdot 10^{-3}$ с;

- згідно з теремою Лармора частота спінової релаксації протонів дорівнює $f_p = \gamma_p B_{\text{вн}}$ [Гц].

7.3. Порядок виконання роботи

Кожний студент для самостійного проведення лабораторної роботи величину зовнішнього магнітного поля має обчислити за формулою:

$$B_{\text{вн}} = (\text{№ зачіткі}) \cdot 10^{-6} \text{ [Тл]},$$

де (№ заліковий) – номер залікової книжки (для прикладу оберемо № = 20477).

Обране в такий спосіб зовнішнє магнітне поле визначить кутову частоту ядерно-магнітного резонансу для обчислювальної моделі в таким чином

$$\omega_m = \frac{2\pi \cdot (\gamma_p \cdot B_{\text{вн}})}{m_t} = 2\pi \cdot (42576375 \cdot 0.020477) \cdot 0.5 \cdot 10^{-3} \approx 2739.$$

Для побудови обчислювальної моделі необхідно вихідну математичну модель перетворити в систему співвідношень, дозволених щодо вихідних змінних, як це зроблене в (7.2). Співвідношення мають виявитися такими, щоб для кожного з них можна було вибрати схему або блок, що обчислює вихідну величину.

7.3.1. Робота в пакеті моделювання

Даний пакет (розділ 1 Micro-Cap v.7-9) за допомогою умовних графічних зображень електричних і електронних компонентів, викликуваних з головного меню, дозволяє побудувати схему обчислювальної моделі. З кожним викликуваним на екран монітора компонентом у машину вводиться аналітичне вираження рівності, що зв'язує ім'я вихідної змінної з іменем вхідної.

Зі змінними пов'язуються поточні значення фізичних явищ, виражені в основній системі електричних одиниць: Вольт, Ампер, секунда. Параметри електричних схем також мають розмірність базових одиниць: Фарада, Генрі, Ом. У зв'язку з цією домовленістю, у записі значень змінних і параметрів аббревіатура назви одиниць виміру не записується. Для вказівки похідних одиниць використовують такі символи: **p** – piko (10^{-12}), **n** – nano (10^{-9}), **u** – micro (10^{-6}), **m** – milli (10^{-3}), **k** – kilo (10^3), **Meg** – mega (10^6), **T** – terra (10^{12}) та ін.

Указавши мишкою в меню на необхідний компонент і клацнувши після

цього на область розміщення компонента на екрані, починайте заповнення властивостей цього компонента у вікні, що сплило при цьому. Виведене на екран зображення можна переміщати, натиснувши на ньому ліву клавішу, і обертати клацаннями правої клавіші, не відпускаючи при цьому ліву.

Іменам об'єктів за допомогою оператора «define <ім'я> <вираз>» можна привласнити або перепривласнити необхідне значення.

Сполучні лінії являють собою електричні провідники, за допомогою яких вихідні електричні величини з одного елемента передаються на входи інших, утворюючи в такий спосіб складну електричну схему на екрані, а в машині – систему рівнянь, яка і є обчислювальною моделлю.

Після побудови обчислювальної схеми її необхідно зберегти й перейти в режим аналізу (тобто розв'язання). Якщо не з'явилося ніяких повідомлень, то з погляду синтаксису схема виявилася несуперечливою і можна переходити до завдання параметрів для незалежної змінної у вікні, що відкрилося.

7.3.2. Порядок побудови схеми для системи рівнянь Блоха

Розглянемо першу рівність (7.2). Щоб обчислити значення похідної \dot{M}_x необхідно одержати алгебраїчну суму двох доданків з негативними знаками. Для цього в переліку макропевних блоків можна відшукати суматор на 3 входи, дати імена k_{ij} цим входам, у яких індекс i – нумерує обчислювальний блок, а індекс j – порядковий номер входу, починаючи, наприклад, зверху. Командами «define k_{ij} ...» будемо задавати їм конкретні значення або нові символічні параметри.

Для обчислення шуканої змінної M_x за її похідною \dot{M}_x можна використовувати макропевний блок інтегратора, з'єднавши вхід останнього з виходом суматора. Продовжуючи аналогічні побудови можна в остаточному підсумку одержати всю обчислювальну модель вихідної системи рівнянь, подібну до тієї, що наведено на рисунку 7.1.

На рисунку 7.1 поряд з позначеннями змінних, параметрів і командами присвоєння, показані позначення, уведені машиною, наприклад: X_1, \dots, X_7 – для блоків і їх параметрів k_{ij} ($i, j = 1, 2, 3$); M_x, \dots, M_z, T_{90} – для міток і схованих з'єднань між вузлами схеми.

У лівій частині для зручного інформування про введені співвідношення й числові дані розміщені команди присвоєння. На схему вони наносяться у режимі тексту (кнопка [T]). Усі об'єкти і написи можуть перетаскуватися лівою клавішею миші в будь-яке місце на екрані.

вектор поперечної складової робить близько двох і одного обороту навколо вертикальної осі, але в різні сторони. На рисунку 7.5 наведена початкова частина графіків перехідних процесів у тій частині, де діє імпульс накачування.

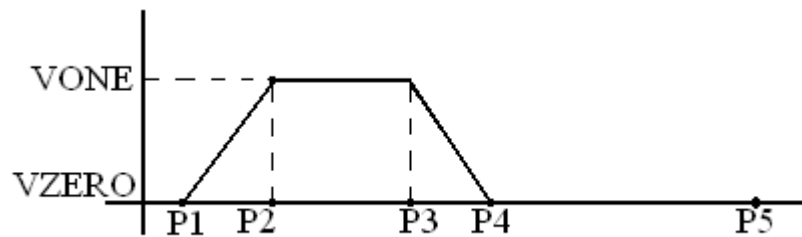


Рисунок 7.2 – Визначення параметрів для імпульсного джерела

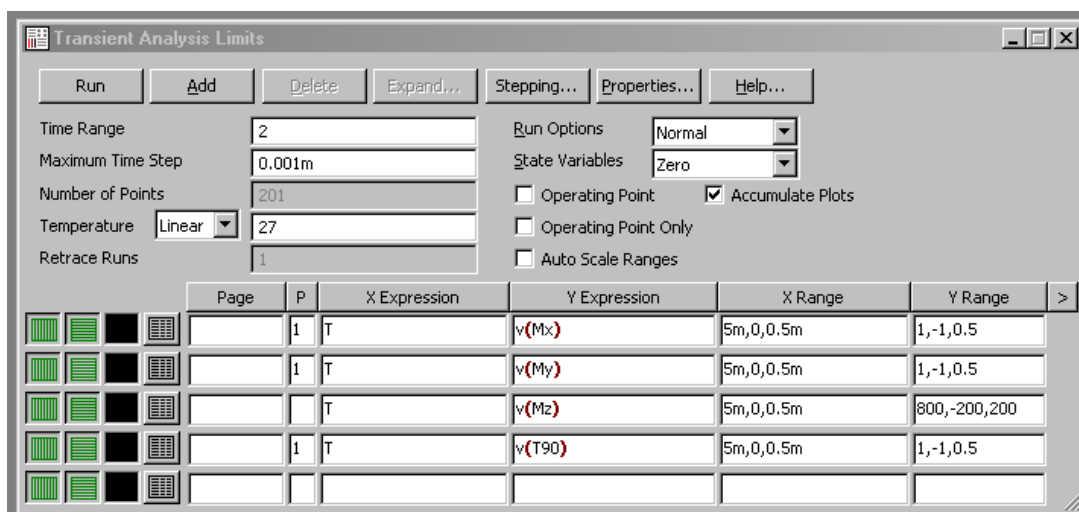


Рисунок 7.3. – Вікно завдання параметрів аналізу

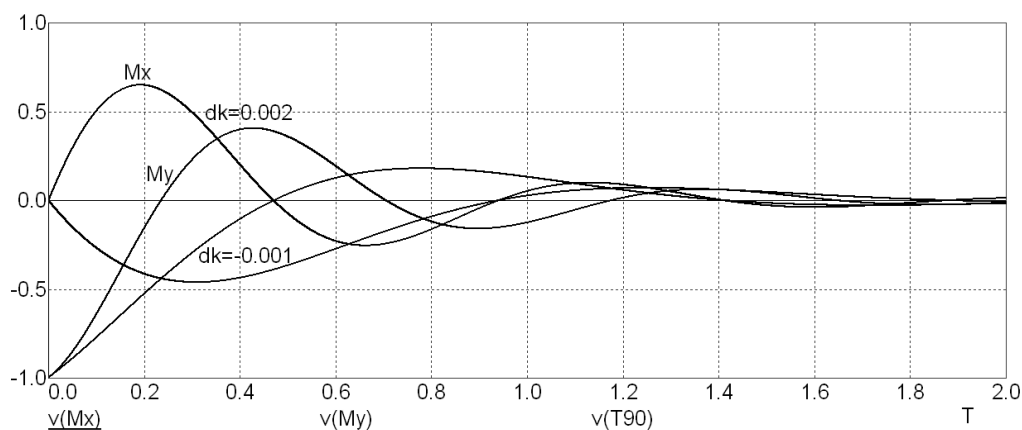


Рисунок 7.4 – Характер обертання Мохам при розстроюваннях різних знаків

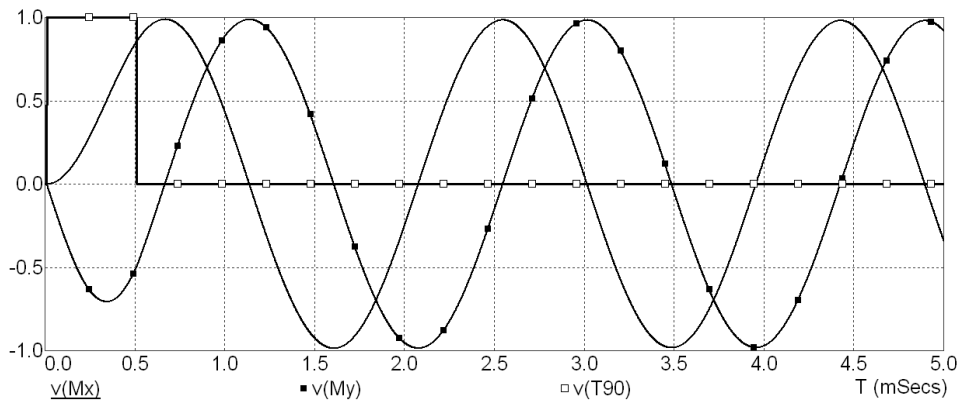


Рисунок 7.5 – Розтягнута початкова частина графіків з рис. 7.4

Можливості пакета моделювання дозволяють міняти параметри обчислювальної моделі в широких межах і цим у даній роботі необхідно скористатися для більш глибокого розуміння процесів ЯМР.

7.4. Завдання на проведення досліджень

1. Побудувати обчислювальну модель рівнянь Блоха.
2. Розрахувати параметри обчислювальної моделі.
3. Зняти графіки зміни поперечних складових для розстроювання в інтервалі $-0.03 < dk < 0.03$.
4. Зняти графіки при різних тривалостях імпульсу накачування й пояснити виникаючі при цьому зміни амплітуд поперечних складових вектора макро намагнічування.
5. Підібрати тривалість імпульсу накачування такої величини, щоб амплітуда поперечної складової по осі y була дорівнює одиниці, а по осі x – близької до нуля. Пояснити, чому це відповідає реальному вектору макро намагнічування.
6. Оформити звіт за проведеними дослідженнями, у якому помістити розрахунки параметрів моделі, графіки перехідних процесів з розмітками характерних утворень на кривих, пояснення одержаних результатів.

8. Лабораторна робота №6. Лінійна фільтрація зображень методом згортки

8.1. Мета роботи

- 1) Ознайомитись з математичною основою лінійної фільтрації.
- 2) Ознайомитися з типовими зображеннями згорткових ядер.
- 3) Створити точкове зображення розміром 51×51 пікселів.
- 4) Розглянути або скласти програму реалізації згортки.
- 5) Виконати фільтрацію створеного зображення, вбудовуючи в програму різні ядра, які забезпечать виділення або підкреслення контурів світлих об'єктів, згладжування або зміни інтенсивності за заданими напрямками і тощо.
- 6) З кожного експерименту з тим або іншим типом згорткового ядра провести аналіз отриманих змін зображення і пов'язати їх зі структурою ядра.

8.2. Теоретичні відомості

Кожне значення i, j -го пікселя вихідного зображення і значення його найближчих сусідів мають множитися на вагові коефіцієнти згорткового ядра, після чого підсумовуватись, усереднюватись і подаватись новим значенням пікселя i, j -го відфільтрованого зображення. Іншими словами, значення сірого для нового пікселя перебуває таким чином

$$G_{\text{new}}(i,j) = (1/N) * \sum (G(i+a, j+b) * H(a, b)), \quad N = \sum (H(a, b)), \quad a, b = \{-1, 0, +1\}.$$

Зверніть увагу! При відкритті пакета Derive зробити початкові установки

[Inputmode:=Word, Casemode:=Sensitive, Timesoperator:=Asterisk,
Notationdigits:=4]

Лінійні згорткові фільтри можна розділити на дві категорії: фільтри верхніх частот і фільтри нижніх частот. Перші підкреслюють різкі зміни градацій яскравості, тобто збільшують контрастність, а другі навпаки, розмивають. До фільтрів верхніх частот відносять градієнти і лапласіани. Згорткові ядра типових фільтрів мають загальноприйняті імена:

- 1) Prewitt Filters (нижче надані векторні набори K_{prw0} і K_{prw1});
- 2) Sobel Filters (нижче надані векторні наборами K_{sob0} і K_{sob1}).

$$K_{prw0} := \begin{bmatrix} [-1, -1, -1; 0, 0, 0; 1, 1, 1], [0, -1, -1; 1, 0, -1; 1, 1, 0], [1, 0, -1; 1, 0, -1; 1, 0, -1], [1, 1, 0; 1, 0, -1; 0, -1, -1]; \\ [1, 1, 1; 0, 0, 0; -1, -1, -1], [0, 1, 1; -1, 0, 1; -1, -1, 0], [-1, 0, 1; -1, 0, 1; -1, 0, 1], [-1, -1, 0; -1, 0, 1; 0, 1, 1] \end{bmatrix}.$$

$$K_{prw0} := \begin{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{bmatrix}$$

Рисунок 8.1 – Ядра фільтра Превіта для висвітлювання границь

$$K_{prw1} := [[-1, -1, -1; 0, 1, 0; 1, 1, 1], [0, -1, -1; 1, 1, -1; 1, 1, 0], [1, 0, -1; 1, 1, -1; 1, 0, -1], [1, 1, 0; 1, 1, -1; 0, -1, -1]; [1, 1, 1; 0, 1, 0; -1, -1, -1], [0, 1, 1; -1, 1, 1; -1, -1, 0], [-1, 0, 1; -1, 1, 1; -1, 0, 1], [-1, -1, 0; -1, 1, 1; 0, 1, 1]].$$

$$K_{prw1} := \begin{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & 1 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{bmatrix}$$

Рисунок 8.2 – Ядра фільтра Превіта для підсвічування границь

$$K_{sob0} := [[-1, -2, -1; 0, 0, 0; 1, 2, 1], [0, -1, -2; 1, 0, -1; 2, 1, 0], [1, 0, -1; 2, 0, -2; 1, 0, -1], [2, 1, 0; 1, 0, -1; 0, -1, -2]; [1, 2, 1; 0, 0, 0; -1, -2, -1], [0, 1, 2; -1, 0, 1; -2, -1, 0], [-1, 0, 1; -2, 0, 2; -1, 0, 1], [-2, -1, 0; -1, 0, 1; 0, 1, 2]]$$

$$K_{sob0} := \begin{bmatrix} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \\ \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \end{bmatrix}$$

Рисунок 8.3 – Вид ядер фільтра Собела для висвітлювання границь

$$K_{sob1} := [[-1, -2, -1; 0, 1, 0; 1, 2, 1], [0, -1, -2; 1, 1, -1; 2, 1, 0], [1, 0, -1; 2, 1, -2; 1, 0, -1], [2, 1, 0; 1, 1, -1; 0, -1, -2]; [1, 2, 1; 0, 1, 0; -1, -2, -1], [0, 1, 2; -1, 1, 1; -2, -1, 0], [-1, 0, 1; -2, 1, 2; -1, 0, 1], [-2, -1, 0; -1, 1, 1; 0, 1, 2]].$$

$$K_{sob1} := \begin{bmatrix} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & -2 \\ 1 & 1 & -1 \\ 2 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 \\ 2 & 1 & -2 \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{bmatrix} \\ \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ -1 & -2 & -1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 \\ -1 & 1 & 1 \\ -2 & -1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -2 & 1 & 2 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \end{bmatrix}$$

Рисунок 8.4 – Вигляд ядер фільтра Собела для підсвічування границь

Імена наборів згорткових ядер, які закінчуються на 0, призначені для висвітлювання границь в одному з 8 напрямків, а ті, що закінчуються на 1 - на фоні вихідного зображення підсвічують границі в обраних напрямках.

До лінійних фільтрів нижніх частот ставляться фільтри, згорткові ядра яких виділяють або підсвічують контури в зображеннях у всіх напрямках:

і1) фільтр із ядрами лапласіана, які представлені в наборі **Klap**, виділять контури, якщо сума зовнішніх коефіцієнтів ядра більше центрального, і підсвітить а якщо ні, то підсвітить;

2) що згладжують фільтри з типовими ядрами **Ksmo**, які призначені для зменшення контрастності на краях і деталях зображених об'єктів;

3) гаусовський фільтр із ядрами в наборі **Kgau** подібний тому, що згладжує, але його ефект розмивання більш м'який

$$K_{lap} := [[0, -1, 0; -1, 4, -1; 0, -1, 0], [0, -1, 0; -1, 5, -1; 0, -1, 0], [0, -1, 0; -1, 6, -1; 0, -1, 0], [-1, -1, -1; -1, 8, -1; -1, -1, -1], [-1, -1, -1; -1, 9, -1; -1, -1, -1], [-1, -1, -1; -1, 10, -1; -1, -1, -1], [-1, -2, -1; -2, 12, -2; -1, -2, -1], [-1, -2, -1; -2, 13, -2; -1, -2, -1]].$$

$$K_{lap} := \begin{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \\ \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ -2 & 13 & -2 \\ -1 & -2 & -1 \end{bmatrix} \end{bmatrix}$$

Рисунок 8.5 – Вид лапласіанових ядер для виділення або підсвічування границь

$$K_{smo} := [[0, 1, 0; 1, 0, 1; 0, 1, 0], [0, 1, 0; 1, 1, 1; 0, 1, 0], [0, 2, 0; 2, 1, 2; 0, 2, 0], [0, 4, 0; 4, 1, 4; 0, 4, 0], [1, 1, 1; 1, 0, 1; 1, 1, 1], [1, 1, 1; 1, 1, 1; 1, 1, 1], [2, 1, 2; 2, 1, 2; 2, 1, 2], [4, 1, 4; 4, 1, 4; 4, 1, 4]]$$

2, 2; 2, 1, 2; 2, 2, 2], [4, 4, 4; 4, 1, 4; 4, 4, 4]].

$$K_{smo} := \begin{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 2 & 0 \\ 2 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 4 & 0 \\ 4 & 1 & 4 \\ 0 & 4 & 0 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 2 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 2 \end{bmatrix} & \begin{bmatrix} 4 & 4 & 4 \\ 4 & 1 & 4 \\ 4 & 4 & 4 \end{bmatrix} \end{bmatrix}$$

Рисунок 8.6 – Вид ядер для зменшення контрастності на краях об’єктів

$K_{gau} := [[0, 1, 0; 1, 2, 1; 0, 1, 0], [0, 1, 0; 1, 4, 1; 0, 1, 0], [1, 1, 1; 1, 2, 1; 1, 1, 1]; [1, 1, 1; 1, 4, 1; 1, 1, 1], [1, 2, 1; 2, 4, 2; 1, 2, 1], [1, 4, 1; 4, 16, 4; 1, 4, 1]]$.

$$K_{gau} := \begin{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{bmatrix} \end{bmatrix}$$

Рисунок 8.7 – Вигляд ядер фільтра Лапласа для м'якого згладжування

8.3. Виконання роботи

8.3.1. Створення робочого зображення для випробування фільтрів

Основою побудови випадкового зображення служить функція $ATAN(4*ABS(SIN(t*a+b))^4)$, де $t = x*SIN(\varphi)+y*COS(\varphi)$; a , b , φ – коефіцієнти, обрані випадковим образом.

Програма $Mi(M_)$ формує вихідне зображення розміром 51×51 пікселів. Значенням кожного пікселя є ціле число з діапазону $[0, 255]$.

```
Mi(M_) :=
PROG(
  a1_ := pi*(1 + RANDOM(12)/6),  a2_ := pi*(1 + RANDOM(11)/6),
  a3_ := pi*(1 + RANDOM(10)/6),
  b1_ := 0.17*pi*RANDOM(7), b2_ := 0.17*pi*RANDOM(6),
  b3_ := 0.17*pi*RANDOM(5),
  d1 := ATAN(4*ABS(SIN(t*a1_ + b1_))^4),
```

```

d2 := ATAN(4*ABS(SIN(t*a2_ + b2_))^4),
d3 := ATAN(4*ABS(SIN(t*a3_ + b3_))^4),
c1_ := LIM(LIM(d1, t, x*SIN(φ) + y*COS(φ), 0), φ, -0.5, 0),
c2_ := LIM(LIM(d2, t, x*SIN(φ) + y*COS(φ), 0), φ, 0.5, 0),
c3_ := LIM(LIM(d3, t, x*SIN(φ) + y*COS(φ), 0), φ, 1.5, 0),
Mz := [], M_ := [], rg := c1_ + c2_ + c3_,
Mz := VECTOR(VECTOR(rg, x, -0.5, 0.5, 0.02), y, -0.5, 0.5, 0.02),
M_ := FLOOR(255*Mz/MAX(MAX(Mz))),
Misx := M_,
RETURN Misx)

```

Щоразу після виконання функції $Mi(M_)$ у пам'яті розміщається і виводиться на поле алгебри зображення квазівипадкового виду з іменем $Misx$. Для відбору своєї картини індивідуально кожному студентові, рекомендується виконати цю програму відразу багато раз підряд, наприклад, увести з командного рядка і виконати, натиснувши на кнопку $[\approx]$, наступний вектор

$$[A:=Mi(M_), B:=Mi(M_), C:=Mi(M_), D:=Mi(M_), \\ E:=Mi(M_), F:=Mi(M_), G:=Mi(M_)].$$

Виведений вектор з сімома матрицями зображень можна відразу ж забрати з екрана, але в пам'яті ці масиви залишаться пов'язаними з іменами $\{A, \dots, G\}$. Створивши на полі алгебри вектор з такими компонентами-іменами, можна, вказуючи на те або інше ім'я, зручно переглянути всі картини $[A, B, C, D, E, F, G]$. Вибравши необхідну картинку дайте їй нове (робоче) ім'я, наприклад $P:=C$.

При вході в графічний режим для перегляду тривимірних зображень установити в меню **Options** режим **Approximate Before Plotting** і настроїти необхідний вид плоского зображення у прямокутній системі координат. У звіт з лабораторної роботи відібране зображення має бути вставлене у вигляді плоскої картини (рис. 8.8).

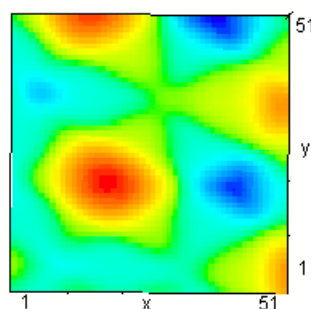


Рисунок 8.8 – Вигляд відібраної випадкової картини для роботи

8.3.2. Побудова програми зашумлення

Для оцінки фільтруючих властивостей необхідно на вихідне зображення накладати шум. Для цього складемо програму $\text{Mrnd}(m)$, яка випадковим чином на полі пікселів розміром 51×51 надає одиночні випадкові значення сірого кольору. Кількість таких піків задавати параметром m . Значення пікселів шумової маски складатимемо за модулем 255 зі значеннями пікселів вихідного зображення, що реалізується нижче оператором Msh .

$\text{Mrnd}(m) :=$

```
PROG(  
  Rnd_ := VECTOR(VECTOR(0, j, 1, 51), i, 1, 51),  
  k := 1,  
  LOOP(  
    r1_ := RANDOM(251), r2_ := RANDOM(251), r3_ := RANDOM(255),  
    i := MOD(k + r1_, 51), j := MOD(k + r2_, 51), Rnd_↓i↓j := r3_,  
    IF(k < m, k := k + 1, RETURN Sh := Rnd_)))
```

Вигляд зашумленого поля Sh , наприклад, з 100 піками можна підібрати так само, як підбиралося вихідне зображення. Надалі його можна викликати за ім'ям Sh або після зашумлення – іншим (рис. 8.9).

Для накладення шумового поля з іменем Sh на поле пікселів зображення з іменем P використовуємо наступний оператор, який необхідно вставити в командний рядок і виконати (результат наведено на рис. 8.10)

$\text{Msh} := \text{VECTOR}(\text{VECTOR}(\text{MOD}(\text{P} \downarrow i \downarrow j + \text{Sh} \downarrow i \downarrow j, 255), j, 1, 51), i, 1, 51)$

8.3.3. Побудова програми згортки

Для реалізації алгоритму згортки виділеної області зображення з ядром фільтра складемо програму з іменем $\text{Newm}(M, H)$, у якій, з метою спрощення її дії на границях зображення, індекси пікселів останнього вибираємо із внутрішньої області значень ($1 < i, j < 51$).

Параметрами програми мають бути матриця (масив) зображення M розміром (51×51) і матриця коефіцієнтів ядра H розміром (3×3) . Програма має вивести на поле Алгебри відфільтрований масив.

$\text{Newm}(M, H) :=$

```
PROG(  
  Mk_ := VECTOR(VECTOR(0, m, 1, 51), n, 1, 51),  
  H_ := VECTOR(H↓(1 + FLOOR(n, 3))↓(1 + MOD(n, 3)), n, 0, 8),  
  nr_ := MAX(Σ(H_↓m, m, 1, 9), 1), i := 2, LOOP( j := 2,  
  LOOP(
```



```

PROG(
  V_ := VECTOR(M↓(i - 1 + FLOOR(n, 3))↓(j - 1 + MOD(n, 3)), n, 0, 8),
  Px_ := FLOOR(1/nr_*V_*H_),
  Pxl_ := IF(Px_ < 0, 0, IF(Px_ > 255, 255, Px_)),
  w_ := IF(((i = 1 OR i = 51) OR j = 1) OR j = 51, 0, Pxl_), Mk_↓i↓j := w_,
  IF(j < 50, j :=+ 1, exit), IF(i < 50, i :=+ 1, exit),
RETURN Mk_)

```

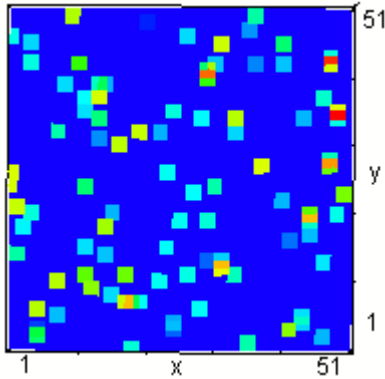


Рисунок 8.9 – Зашумлене поле

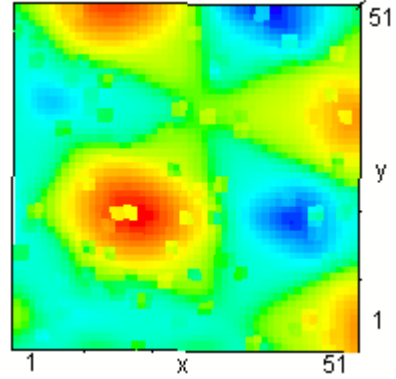


Рисунок 8.10 – Зашумлене зображення

Послідовність дії операторів у програмі.

Спочатку, перед вкладеними друг у друга циклами, скинути в нуль робочий масив $Mk_$ розміром 51×51 . Потім перетворити в одновірний вектор $H_$ матриця ядра фільтра і обчислити його норму $nr_$. У внутрішньому циклі кожній парі поточних індексів i, j створюється одновірний вектор значень навколишніх ачумлено $V_$, відповідний до структури і розміру вектора ядра, обчислюється скалярний добуток векторів і отримана в результаті сума покомпонентних добутоків усереднюється шляхом ділення на коефіцієнт, що нормує. Результатом цих дій є нове значення i, j -того пікселя. Для граничних значень індексів масиву зображення оператором $w_$ граничним ачумлено примусово встановлюється значення нуль.

Для прикладу, згортчим алгоритмом виконаємо фільтрацію ачумленого зображення Msh з ядром із набору $Kprw0$. Програму виконаємо відразу для 4-х ядер за допомогою наступного вектора, у яким імена обрані відповідно до індексів ядер у наборі $Kprw0$:

```

[M11:=Newm(Msh, Kprw0↓1↓1),M13:=Newm(Msh,
Kprw0↓1↓3),M22:=Newm(Msh, Kprw0↓2↓2),M24:=Newm(Msh, Kprw0↓2↓4)].

```

Перегляд і запам'ятовування картинок виконаємо з вектора з компонентами-іменами такого вигляду $[M11, M13, M22, M24]$.

Вплив кожного фільтра на зображення наведено на рисунку 8.11.

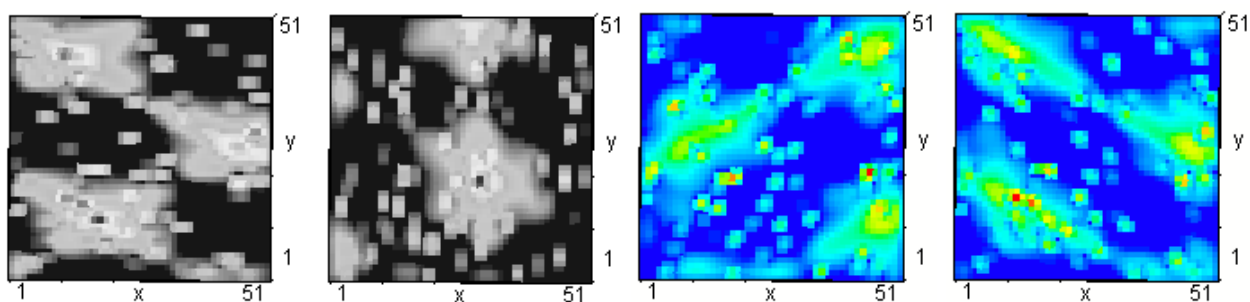


Рисунок 8.11 – Результат дії фільтрів (ліворуч-праворуч) M11,M13,M22,M24

Зрівняємо тепер дію фільтруючих ядер з набору Kprw1 для тих же індексів. Для цього сформуємо аналогічно попередньому такі вектори:

[P11:=Newm(Msh, Kprw1↓1↓1),P13:=Newm(Msh, Kprw1↓1↓3),
P22:=Newm(Msh, Kprw1↓2↓2),P24:=Newm(Msh, Kprw1↓2↓4)]
[P11,P13,P22,P24].

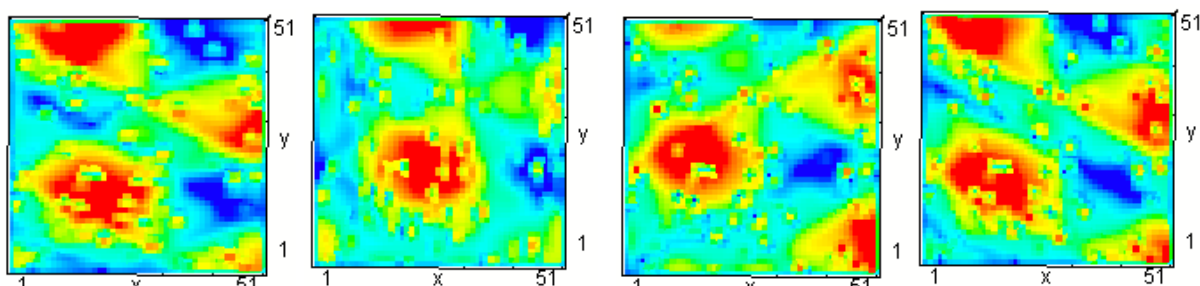


Рисунок 8.12 – Результат дії фільтрів (ліворуч-праворуч) P11, P13, P22, P24

Аналіз картинок на рисунках 8.10-8.12 показує ту особливість ядер, що фільтри верхніх частот типу Kprw0 виділяють границі об'єктів, притіняючи все зображення, а фільтри типу Kprw1 підсвічують їх, не зачіпаючи об'єктів у цілому. Самостійно до зашумленого масиву, зображеного на рис. 8.10, використовуйте вибірково ядра фільтрів Ksob0, Ksob1, Klap, Ksmo, Kgau.

8.4. Хід виконання роботи

Познайомитися з ідеєю фільтрації та способом вистави згорткових ядер. Детально познайомитися із програмами формування масиву зображення, масиву зашумлення, зашумленого зображення та програмою, що виконує власне фільтрацію. У викладача отримайте індивідуальний перелік згорткових ядер, з якими будете проводити конкретні дослідження або, принаймні, виберіть із кожного набору ядер по 4 ядра.

Звіт з лабораторній роботи має містити пояснення з кожного експерименту і зображення результату з текстом, що підтверджують досягнення мети фільтрації. Мають бути наведені та описані самостійні дії з каскадної фільтрації зображення.

9. Лабораторна робота №7. Нелінійна фільтрація зображень

9.1. Мета роботи

1. Ознайомитися з математичною основою нелінійної фільтрації.
2. Ознайомитися з типовими нелінійними співвідношеннями для ядер.
3. Створити точкове зображення розміром 51×51 пікселів.
4. Розглянути, скласти або модифікувати програми, які перетворюють значення будь-якого пікселя відповідної до необхідних правил.
5. Виконати фільтрацію створеного зображення, вбудовуючи в ці програми різні нелінійні співвідношення, які забезпечать необхідне перетворення.
6. З кожного експерименту за заданим типом правил перетворення провести аналіз отриманих змін зображення.

9.2. Теоретичні відомості

Нелінійна фільтрація полягає в тому, що кожне значення i, j -го пікселя вихідного зображення замінюється значенням якоїсь нелінійної функції від значень навколишніх пікселів.

Як нелінійні функції найчастіше вибирають такі:

- вибір максимального значення з абсолютних значень кінцевих різниць, обчислених за заданим напрямком (фільтри градієнтні, Робертса, різницеві, та ін.);
- порівняння різниці значення i, j -того пікселя й середнього значення навколишніх пікселів із середньоквадратичним відхиленням (Σ -фільтри – фільтри високих і низьких просторових частот);
- порівняння зваженого значення i, j -того пікселя зі зваженою сумою значень навколишніх пікселів (фільтри медіанний і n -го порядку).

У даній лабораторній роботі використовується вікно, що оточує i, j -й піксель, яке подається як матриця 3×3 . У програмах її зручно подавати вектором VM , отриманим конкатенацією рядків матриці M :

$$VM := [a, b, c, d, e, f, g, h, q],$$

де $[a, b, c] = [1, 2, 3]$ – верхній рядок матриці M та індекси її у векторі;
 $[d, e, f] = [4, 5, 6]$ – середній рядок матриці M та індекси її у векторі;
 $[g, h, q] = [7, 8, 9]$ – нижній рядок матриці M та індекси її у векторі.

Подібним чином зручно подавати і вагові коефіцієнти ядра у вигляді

матриці і/або одновимірного вектора.

Відповідність пікселів $p(i, j)$ матриці M , що оточують точку (i, j) , елементам її вектора VM і коефіцієнтам ядра Kk наведена нижче

$$\begin{bmatrix} p(i-1, j-1) & p(i, j-1) & p(i+1, j-1) \\ p(i-1, j) & p(i, j) & p(i+1, j) \\ p(i-1, j+1) & p(i, j+1) & p(i+1, j+1) \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & q \end{bmatrix} = \begin{bmatrix} k(-1, -1) & k(0, -1) & k(+1, -1) \\ k(-1, 0) & k(0, 0) & k(+1, 0) \\ k(-1, +1) & k(0, +1) & k(+1, +1) \end{bmatrix}$$

Якщо пікселі з вікна зображені як вектор Vp , а коефіцієнти ядра – вектор Kk , то розраховану суму значень пікселів у вікні можна обчислити, використовуючи їх скалярний добуток

$$(Vp, Kk) = \sum_{m=1}^9 (Vp(m) \cdot Kk(m)).$$

Якщо необхідні тільки значення навколишніх пікселів, то зі скалярного добутку слід відняти добуток центральних елементів

$$(Vp, Kk) - Vp(5) \cdot Kk(5).$$

Використовуючи скалярний добуток векторів, зручно також обчислювати стандартне (середньоквадратичне) відхилення суми значень пікселів:

$$S = \sqrt{(Vp, Kk)}.$$

9.3. Види нелінійних фільтрів з вікном 3×3

При вході в Derive зробіть основні початкові установки:

[Inputmode:=Word, Casemode:=Sensitive, Timesoperator:=Asterisk, Notationdigits:=4].

9.3.1. Нелінійний фільтр Превіта

Нелінійний фільтр Превіта забражуються ядрами $Kprwn$, що обчислюють за значеннями пікселів кінцеві різниці в декількох напрямках, і нелінійну функції, що вибирає з них максимальне значення.

Вигляд матриць коефіцієнтів у командному рядку записується так:

$$\begin{aligned} \text{Kprw} := & [[-1, -1, -1; 0, 0, 0; 1, 1, 1], [-1, 0, 1; -1, 0, 1; -1, 0, 1], \\ & [0, 1, 1; -1, 1, 1; -1, -1, 0], [1, 1, 0; 1, 1, -1; 0, -1, -1]]. \end{aligned}$$

Кожне, наприклад, n -е ядро із запису вектора Kprw вибирається за індексом

$$\text{Kprw} \downarrow n.$$

Конкатенацію рядків для двох перших ядер вектора Kprw можна одержати, принаймні, двома різними операторами:

$$\begin{aligned} \text{J1} &:= \text{APPEND}(\text{VECTOR}(\text{Kprw} \downarrow 1 \downarrow i, i, 1, 3)), \\ \text{J1} &:= [-1, -1, -1, 0, 0, 0, 1, 1, 1]; \\ \text{J2} &:= \text{VECTOR}(\text{Kprw} \downarrow 2 \downarrow (1 + \text{FLOOR}(n, 3)) \downarrow (1 + \text{MOD}(n, 3)), n, 0, 8), \\ \text{J2} &:= [-1, 0, 1, -1, 0, 1, -1, 0, 1]. \\ (\text{VM} * \text{J1}) &\Rightarrow (-a - b - c + g + h + q), \\ (\text{VM} * \text{J2}) &\Rightarrow (-a + c - d + f - g + q). \end{aligned}$$

Функція вибору для нового значення пікселя в нелінійному фільтрі Превіта представляється функцією

$$\text{Fpr} := \text{MAX}(\text{VM} * \text{J1}, \text{VM} * \text{J2}).$$

9.3.2. Нелінійний фільтр Собелу

Нелінійний фільтр Собелу подається ядрами Ksobn , що обчислюють за значеннями пікселів зважені кінцеві різниці в декількох напрямках, і нелінійної функції, що вибирає з них максимальне значення

$$\begin{aligned} \text{Ksob} := & [[-1, -2, -1; 0, 0, 0; 1, 2, 1], [-1, 0, 1; -2, 0, 2; -1, 0, 1], \\ & [0, 1, 2; -1, 1, 1; -2, -1, 0], [2, 1, 0; 1, 1, -1; 0, -1, -2]]. \end{aligned}$$

Якщо вектори двох ядер, наприклад, першого і другого є

$$\begin{aligned} \text{Js1} &:= [-1, -2, -1, 0, 0, 0, 1, 2, 1], \\ \text{Js2} &:= [-1, 0, 1, -2, 0, 2, -1, 0, 1], \end{aligned}$$

то функція вибору для нового значення пікселя в нелінійному фільтрі Собелу подається функцією

$$\begin{aligned} \text{Fsb} &:= \text{MAX}([\text{ABS}(\text{VM} * \text{Js1}), \text{ABS}(\text{VM} * \text{Js2})]) \text{ або} \\ \text{Fsb} &:= \text{MAX}([\text{ABS}(a + 2*b + c - g - 2*h - q), \text{ABS}(a - c + 2*d - 2*f + g - q)]). \end{aligned}$$

9.3.3. Градієнтний фільтр

Nlgra := [[-1, 0, 0; 1, 0, 0; 0, 0, 0], [0, -1, 0; 0, 1, 0; 0, 0, 0]];

Fgr := MAX([ABS(VM*Jg1), ABS(VM*Jg2)]) або

Fgr := MAX([ABS(a – d), ABS(b – e)]).

9.3.4. Фільтр Робертса

Nlrob := [[-1, 0, 0; 0, 1, 0; 0, 0, 0], [0, -1, 0; 1, 0, 0; 0, 0, 0]];

Frb := MAX([ABS(VM*Jr1), ABS(VM*Jr2)]) або

Frb := MAX([ABS(a – e), ABS(b – d)]).

9.3.5. Різницевий фільтр із вибіркою за трьома напрямками

Nlrz := [[-1, 0, 0; 0, 1, 0; 0, 0, 0], [0, 0, 0; -1, 1, 0; 0, 0, 0],

[0, -1, 0; 0, 1, 0; 0, 0, 0]];

Frz:=MAX([ABS(VM*Jz1), ABS(VM*Jz2), ABS(VM*Jz3)]) або

Frz:=MAX([ABS(a – e), ABS(d – e), ABS(b – e)]).

9.3.6. Сигма фільтри

До сигма фільтрів належать фільтри, у яких різниця значення пікселя $p(i, j)$ дорівнює середньому значенню

$$M = \frac{1}{9} \cdot \sum_{i,j \in \{1,2,3\}} p(i, j)$$

пікселів вікна зі стандартним середньоквадратичним відхиленням пікселів вікна

$$S = \sqrt{\frac{1}{9} \cdot \sum_{i,j \in \{1,2,3\}} (p(i, j) - M)^2}.$$

У високочастотному сігма-фільтрі підкреслюється перебільшення різниці над стандартним відхиленням, а в низькочастотному – навпаки:

Для фільтра
верхніх частот

if $p(i, j - M) > S$
then $p(i, j) := p(i, j)$
else $p(i, j) := M;$

Для фільтра
нижніх частот

if $p(i, j - M) < S$
then $p(i, j) := p(i, j)$
else $p(i, j) := M;$

9.3.7. Фільтри з різними ваговими коефіцієнтами вікна

Медіанний фільтр має у вікні всі коефіцієнти, що дорівнюють одиниці. У загальному випадку це фільтри n -го порядку, які здійснюють зважене підсумовування значень навколишніх пікселів.

9.4. Вихідне зображення та зашумлення

Методику використання і програму побудови вихідного зображення, а також програму зашумлення зображення беремо з лабораторної роботи №4.

9.4.1. Програма вибору зображення

Текст програми $Mi(M_)$ перебуває в розділі 8.3.1, який звідти необхідно скопіювати у багаторядковий командний рядок і на поле **Алгебри**. Для відбирання зображення введіть у такий же спосіб на поле **Алгебри** наступні два вектори:

$[A:=Mi(M_), B:=Mi(M_), C:=Mi(M_), D:=Mi(M_), E:=Mi(M_),$
 $F:=Mi(M_), G:=Mi(M_)], \quad [A, B, \blacksquare, D, E, F, G].$

Відібрану матрицю також назвемо іменем P .

9.4.2. Програма вибору матриці зашумлення

Текст програми $Mrnd(m)$ перебуває в розділі 8.3.2, який звідти необхідно скопіювати у багаторядковий командний рядок і на поле **Алгебри**.

Виконаємо програму зашумлення з параметром $m=100$. При цьому масив зашумлення формується з іменем Sh .

Оператор накладення матриці шуму на матрицю зображення

$Msh := VECTOR(VECTOR(MOD(P \downarrow i \downarrow j + Sh \downarrow i \downarrow j, 255), j, 1, 51), i, 1, 51)$

Вихідна і зашумлена матриця після виконання відповідних програм у розділі 8.3 передані через системний буфер у робочу сесію даної лабораторної роботи і забражені на рис. 9.1.

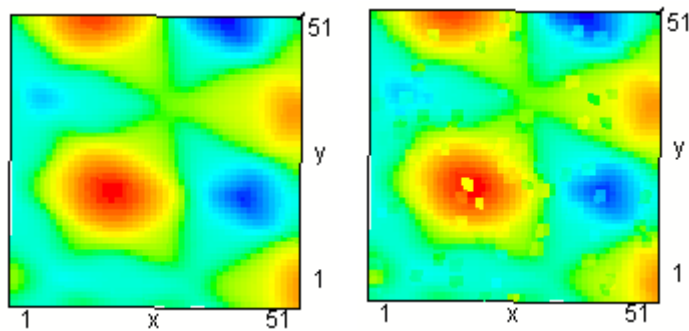


Рисунок 9.1 – Вихідне зображення P і його зашумлення Msh

9.5. Побудова програми нелінійної фільтрації

Програма нелінійної фільтрації $Nonl(M, H1, H2)$ призначає для виділення або підкреслення границь об'єктів відповідно до ядер, що виділяють $H1$ і $H2$ з розділів 9.3.1–9.3.5.

$Nonl(M, H1, H2) :=$

```

PROG(
  Mk_ := VECTOR(VECTOR(0, m, 1, 51), n, 1, 51),
  H1_ := VECTOR(H1↓(1 + FLOOR(n, 3))↓(1 + MOD(n, 3)), n, 0, 8),
  H2_ := VECTOR(H2↓(1 + FLOOR(n, 3))↓(1 + MOD(n, 3)), n, 0, 8),
  i := 2,
  LOOP( j := 2,
    LOOP(
      PROG(
        V_ := VECTOR(M↓(i - 1 + FLOOR(n, 3))↓(j - 1 + MOD(n, 3)), n, 0, 8),
        u1_ := VECTOR(V_↓k*H1_↓k, k, 1, 9),
        u2_ := VECTOR(V_↓k*H2_↓k, k, 1, 9),
        s1_ := Σ(u1_↓k, k, [1, 2, 3, 4, 6, 7, 8, 9]),
        s2_ := Σ(u2_↓k, k, [1, 2, 3, 4, 6, 7, 8, 9]),
        px := FLOOR(MAX(ABS(s1_), ABS(s2_))),
        pxl_ := IF(px < 0, 0, IF(px > 255, 255, px)),
        w_ := IF(((i = 1 OR i = 51) OR j = 1) OR j = 51, 0, pxl_),
        Mk_↓i↓j := w_,
        IF(j < 50, j :=+ 1, exit) ),
      IF(i < 50, i :=+ 1, exit) ),
    RETURN Mk_)

```

Наприклад пропустимо незашумлене P і зашумлене зображення Msh відповідно через фільтри Превіта, Собелу та Робертса, послідовно набравши в

командному рядку і виконавши зазначені нижче функції. Результати роботи наведені на рис. 9.2.

Nonl(P, Kprw↓1, Kprw↓2), Nonl(P, Ksob↓1, Ksob↓2),
 Nonl(P, Nlrob↓1, Nlrob↓2).
 Nonl(Msh, Kprw↓1, Kprw↓2), Nonl(Msh, Ksob↓1, Ksob↓2),
 Nonl(Msh, Nlrob↓1, Nlrob↓2)

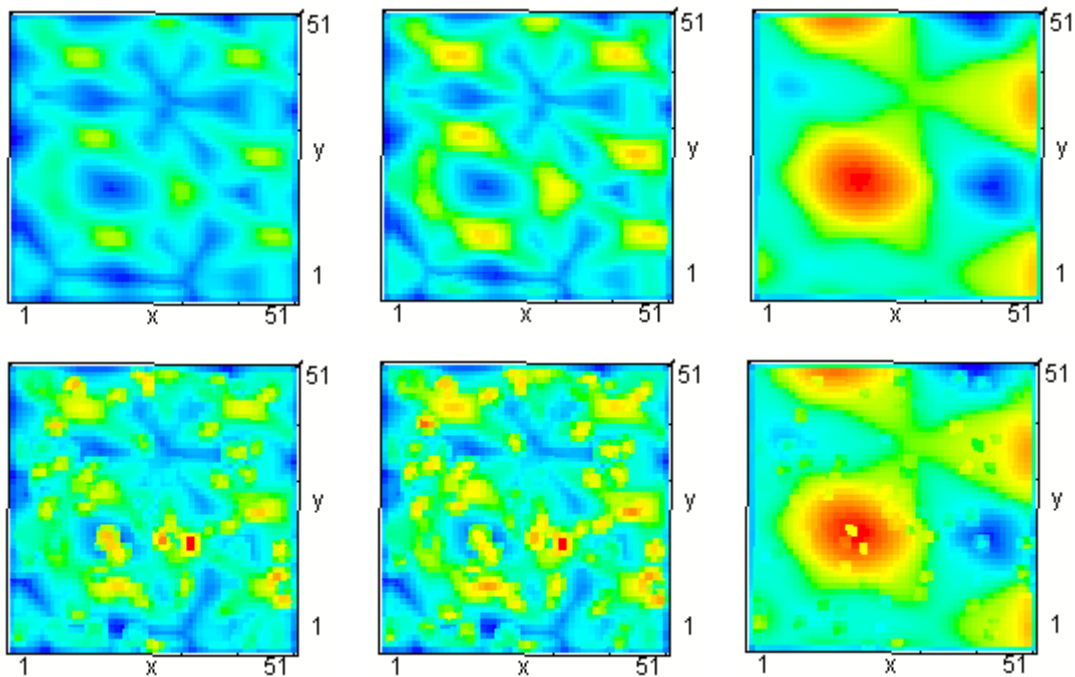


Рисунок 9.2 – Робота з ядрами 1 і 2 фільтрів Превіта, Собелу та Робертса

Можна відзначити, що фільтри, які використовують різниці, часто знижують контрастність зображення. Її можна підвищити за допомогою програми, де кожний піксель ділиться на значення самого яскравого пікселя в зображенні і результат помножиться на 255. Нижче подається текст такої програми.

```
Kontrast(Mk_) :=
  PROG(
    Mjob := VECTOR(VECTOR(0, j, 1, 51), i, 1, 51),
    mx_ := MAX(MAX(Mk_)), i := 1,
    LOOP( j := 1,
      LOOP( PROG(
        px_ := Mk_↓j↓i, pxn_ := FLOOR(px_/mx_*255),
        pxres_ := IF(pxn_ < 0, 0, IF(pxn_ > 255, 255, pxn_)),
        Mjob↓j↓i := pxres_, IF(j < 51, j := j + 1, exit) ),
```

```

IF(i < 51, i :+ 1, exit) ),
RETURN Mkont := Mjob)

```

Програму підвищення контрастності випробуємо на лівій зверху картинці, наведеної на рис. 9.2, після роботи фільтра $\text{Nonl}(P, \text{Kprw} \downarrow 1, \text{Kprw} \downarrow 2)$.

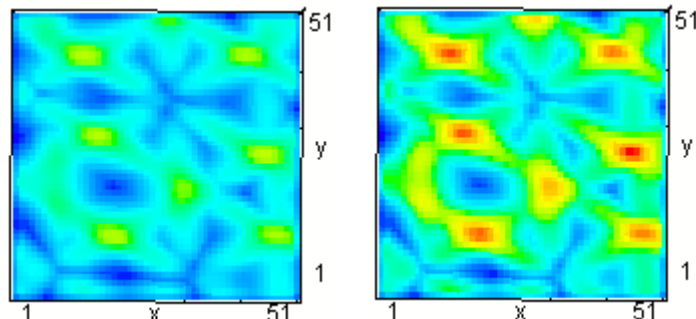


Рисунок 9.3 – Робота програми підвищення контрастності

Для вибору максимальної корекції за трьома різницеви́ми ядрами у програму нелінійної фільтрації необхідно внести відповідні доповнення. Текст видозміненої програми $\text{Nlrazn}(M, H1, H2, H3)$ поданий у такому вигляді.

```

Nlrazn(M, H1, H2, H3) :=
  PROG(
    Mk_ := VECTOR(VECTOR(0, m, 1, 51), n, 1, 51),
    H1_ := VECTOR(H1↓(1 + FLOOR(n, 3))↓(1 + MOD(n, 3)), n, 0, 8),
    H2_ := VECTOR(H2↓(1 + FLOOR(n, 3))↓(1 + MOD(n, 3)), n, 0, 8),
    H3_ := VECTOR(H2↓(1 + FLOOR(n, 3))↓(1 + MOD(n, 3)), n, 0, 8), i := 2,
    LOOP( j := 2,
      LOOP(
        PROG(
          V_ := VECTOR(M↓(i - 1 + FLOOR(n, 3))↓(j - 1 + MOD(n, 3)), n, 0, 8),
          u1_ := VECTOR(V_↓k*H1_↓k, k, 1, 9),
          u2_ := VECTOR(V_↓k*H2_↓k, k, 1, 9),
          u3_ := VECTOR(V_↓k*H2_↓k, k, 1, 9),
          s1_ := Σ(u1_↓k, k, [1, 2, 3, 4, 6, 7, 8, 9]),
          s2_ := Σ(u2_↓k, k, [1, 2, 3, 4, 6, 7, 8, 9]),
          s3_ := Σ(u2_↓k, k, [1, 2, 3, 4, 6, 7, 8, 9]),
          px := FLOOR(MAX(ABS(s1_), ABS(s2_), ABS(s3_))),
          pxl_ := IF(px < 0, 0, IF(px > 255, 255, px)),
          w_ := IF(((i = 1 OR i = 51) OR j = 1) OR j = 51, 0, pxl_), Mk_↓i↓j := w_,
          IF(j < 50, j :+ 1, exit) ),

```

```

IF(i < 50, i :+ 1, exit) ),
RETURN Mk_).

```

Nlrazn(P, Nlraz↓1, Nlraz↓2, Nlraz↓3)

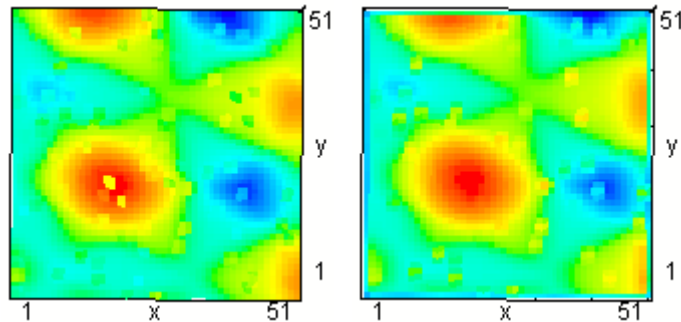


Рисунок 9.4 – Ліворуч зашумлена картинка, а праворуч – після фільтра Nlraz

9.6. Сигма фільтри

У процедурах фільтрації сігма-фільтрів використовуються функції обчислення зважених середніх і середньоквадратичних відхилень. Алгоритми модифікації значення кожного пікселя наведені в розділі 9.3.

9.6.1. Програма сігма-фільтрації верхніх частот

```

Nwsh(M) :=
PROG(
Mk_ := VECTOR(VECTOR(0, m, 1, 51), n, 1, 51),
i := 2,
LOOP(
j := 2,
LOOP(
PROG(
V_ := VECTOR(M↓(i - 1 + FLOOR(n, 3))↓(j - 1 + MOD(n, 3)), n, 0, 8),
MO_ := 1/9*Σ(V_↓k, k, 1, 9),
SQ_ := √(1/9*Σ((MO_ - V_↓k)^2, k, 1, 9)),
a_ := ABS(V_↓5 - MO_),
b_ := V_↓5, c_ := MO_, d_ := SQ_, vv_ := IF(a_ > d_, b_, c_),
vvv_ := IF(vv_ < 0, 0, IF(vv_ > 255, 255, vv_)),
vvvv_ := IF(((i = 1 OR i = 51) OR j = 1) OR j = 51, vvv_ := 0, vvv_)),
Mk_↓i↓j := FLOOR(vvvv_),
IF(j < 50, j :+ 1, exit) ),
IF(i < 50, i :+ 1, exit) ),
RETURN Mk_).

```

9.6.2. Програма сігма-фільтрації нижніх частот

```
Nwsl(M) :=  
  PROG(  
    Mk_ := VECTOR(VECTOR(0, m, 1, 51), n, 1, 51),  
    i := 2,  
    LOOP(  
      j := 2,  
      LOOP(  
        PROG(  
          V_ := VECTOR(M↓(i – 1 + FLOOR(n, 3))↓(j – 1 + MOD(n, 3)), n, 0, 8),  
          MO_ := 1/9*Σ(V_↓k, k, 1, 9),  
          SQ_ := √(1/9*Σ((MO_ – V_↓k)^2, k, 1, 9)),  
          a_ := ABS(V_↓5 – MO_), b_ := V_↓5, c_ := MO_, d_ := SQ_,  
          vv_ := IF(a_ < d_, b_, c_),  
          vvv_ := IF(vv_ < 0, 0, IF(vv_ > 255, 255, vv_)),  
          vvvv_ := IF(((i = 1 OR i = 51) OR j = 1) OR j = 51, vvv_ := 0, vvv_)),  
          Mk_↓i↓j := FLOOR(vvvv_),  
          IF(j < 50, j := j + 1, exit) ),  
          IF(i < 50, i := i + 1, exit) ), RETURN Mk_).
```

9.6.3. Програма медіанного фільтра

Програма медіанного фільтра Medf(M) заміняє кожний піксель зображення середнім значенням з навколишніх його пікселів.

```
Medf(M) :=  
  PROG(  
    Mk_ := VECTOR(VECTOR(0, m, 1, 51), n, 1, 51), i := 2,  
    LOOP(  
      j := 2,  
      LOOP(  
        PROG(  
          V_ := VECTOR(M↓(i – 1 + FLOOR(n, 3))↓(j – 1 + MOD(n, 3)), n, 0, 8),  
          MO_ := 1/8*(Σ(V_↓k, k, 1, 9) – V_↓5), a_ := MO_,  
          v_ := IF(a_ < 0, 0, IF(a_ > 255, 255, a_)),  
          vv_ := IF(((i = 1 OR i = 51) OR j = 1) OR j = 51, v_ := 0, v_)),  
          Mk_↓i↓j := FLOOR(vv_),  
          IF(j < 50, j := j + 1, exit) ), IF(i < 50, i := i + 1, exit) ),  
    RETURN Mk_)
```

Самостійно модифікуйте або розробіть заново програму, яка буде виконувати фільтрацію n -го порядку, і застосуйте її до вихідного та зашумленого зображення. Випробуйте каскадне виконання фільтрації декількома фільтрами.

За необхідності використовуйте програму підвищення контрастності.

З кожного експерименту зробіть оцінні висновки.

9.7. Хід виконання роботи

Уважно ознайомтеся з ідеями нелінійної фільтрації. Сформууйте масив зображення, масив зашумлення, зашумлене зображення.

У викладача отримайте індивідуальний перелік програм, з якими будете проводити конкретні дослідження.

Звіт з лабораторної роботи має містити в собі пояснення з кожного експерименту і зображення результату з текстом, що підтверджують досягнення мети фільтрації.

Мають бути наведені і описані самостійні дії з каскадної фільтрації зображення. Результати підтверджуються картинками зображень.

10. Лабораторна робота № 8. Розкладання дискретних випадкових векторів у ряд Карунена-Лоева за власними векторами кореляційної матриці

10.1. Мета роботи

Ознайомитися з методикою розкладання випадкових векторів за власними векторами кореляційної матриці, сформованої із множини зображень одного виду. Познайтися із процедурами побудови кореляційної матриці, обчислення власних значень матриці, власних векторів і відповідно до розкладання вихідного вектора зображення по проєкціях на власні вектори кореляційної матриці.

10.2. Теоретичні відомості

Кожне зображення можна подати багатомірним вектором, розмірність якого формально дорівнює числу пікселів. Лінійна впорядкованість пікселів по компонентах такого вектора визначається способом розгорнення зображення на плоскому екрані монітора. Метою стиснення такого вектора є зменшення кількості елементів зі збереженням найважливіших ознак, за якими вид зображення, хоча і з деякими викривленнями, можна відновити.

Ідея стиснення n -вимірною вектора методом Карунена-Лоева (КЛ) полягає в тому, щоб вибрати і використовувати таку n -вимірну систему ортогональних вісей, у якій значимі проєкції вихідного вектора зображення виявилися б лише на k осях координатної системи ($n > k$), а значення інших проєкцій можна було б зробити нульовими або мізерно малими. Якщо повернути зворотним лінійним перетворенням новий вектор у вихідну систему координат, то буде отриманий зразок вихідного зображення з певним ступенем правдоподібності.

Будь-яке зображення можна розглядати як корисну інформаційну частину, на яку накладена адитивна шумова складова. Виділення корисної складової із зашумленого ряду випадкових чисел, який є послідовністю пікселів у розгорненні зображення, дозволяє використовувати кореляційні методи, особливо якщо корисна складова має гаусовське розподілення.

10.3. Послідовність виконання роботи

1. Згенерувати кілька зображень одного виду, за якими побудувати кореляційну матрицю.
2. Знайти власні значення кореляційної матриці та зробити їх

рандомізацію за значущістю.

3. Установивши критерій значущості, відібрати власні значення матриці, для яких обчислити власні вектори.

4. За власними векторами виконати ортнормування матриці, яка дозволить подавати вектор чергового зображення в проекціях на власні вектори деякої еталонної кореляційної матриці для зображень заданого виду.

10.4. Приклад виконання лабораторної роботи

Індивідуалізація завдання для кожного студента здійснюється випадковим набором зображень одного класу, які послужать побудові кореляційної матриці.

Як базове зображення у даній роботі пропонується обрати зображення з лабораторної роботи №6 (Розділ 8.3), що має розміри 51×51 пікселів. Довжина вектора, сформованого шляхом розгорнення цієї матриці, більш ніж 2500. Крім того, насиченість дрібними деталями зображень звичайно невелика й масиву, запропонованого для роботи також. Тому велика кількість подібних зображень ми одержимо шляхом багаторазового застосування спрощеного розгорнення вихідного зображення. Спрощене розгорнення оберемо таким, щоб псевдо зображення мало розмір матриці з пікселів 16×16 . Зміст розгорнення полягає в тому, що з деякого випадково обраного початкового пікселя, розташованого в межах квадратної області 4×4 у лівому верхньому куті зображення, з кожного рядка послідовно із кроком 15 по горизонталі й вертикалі вибираються пікселі. У такий спосіб з вихідного зображення 51×51 будуть сформовано 16 зображень 4×4 , які будуть представлені 16-ті вимірними випадковими векторами.

10.4.1. Одержання масиву векторів з вихідного зображення

Відповідно до пункту 8.3.1 за програмою $M_i(M_)$ створюємо масив зображення 51×51 з іменем M_p . Можливий вигляд його наведений на рис. 8.8.

Формування одного варіанта спрощеного зображення й вистава його вектором можна виконати наступними двома вкладеними операторами:

$$\begin{aligned} X &:= \text{VECTOR}(\text{VECTOR}(M_p \downarrow i \downarrow j, j, \text{RANDOM}(4) + 1, 51, 15), i, \text{RANDOM}(4) + \\ &\quad 1, 51, 15) \\ V &:= \text{APPEND}(\text{VECTOR}(X \downarrow m, m, 1, 4)) \end{aligned}$$

У першому для сканування задається випадкова початкова точка, а в другому спрощена матриця X розміру 4 (4 подається 16-і вимірним вектором. Завдяки використанню процедури генерування випадкових чисел, при кожному запуску на виконання другого вираження буде скановано інше спрощене

зображення і отримано його векторний вигляд. Для багаторазових запусків і формування масиву **R** з 16 однотипними спрощеними випадковими 16-ті вимірних векторів можна скористатися такою циклічною програмою, ядром якої є оператор APPEND, що виконує конкатенацію матриць [V] з розміром (1(16)):

$$R:=\text{PROG}(M:=[V],k:=1,\text{LOOP}(M:=\text{APPEND}(M,[V]), \\ \text{IF}(k\geq 16,\text{RETURN } R:=M,k:=k+1)))$$

Для цього базового набору випадкових векторів будемо будувати ортогональну координатну систему із власних векторів кореляційної матриці.

10.4.2. Побудова кореляційної матриці

Насамперед необхідно виконати центрування випадкових векторів зображення. Для цього в кожному вхідному векторі слід знайти середнє значення і відняти його від кожного компонента.

Вектор середніх значень для всіх 16 векторів зображень

$$m_i:=\text{vector}(1/16*\sum(R_{\downarrow i\downarrow j},i,1,16),j,1,16)$$

Центрована матриця вихідних 16 зображень

$$M_z:=\text{vector}(\text{VECTOR}(R_{\downarrow i\downarrow j}-m_{i\downarrow j},i,1,16),j,1,16)'$$

Після виклику оператора **Mz**, отриманий масив центрованих векторів з екрана монітора можна вилучити. У пам'яті він залишається до кінця сесії. Його значення можна вивести на екран знову, викликавши функцію за ім'ям **Mz**.

Обчислюємо середньоквадратичне відхилення компонентів у кожного випадкового центрованого вектора, представивши їх вектором для всієї їхньої сукупності

$$\sigma_i:=\text{vector}(\sqrt{(1/(\text{DIMENSION}(M_z')-1)*M_z_{\downarrow i}*M_z_{\downarrow i})},i,1,16)$$

Кореляційну матрицю з нормованих коефіцієнтів взаємної кореляції обчислимо в такий спосіб

$$K:=\text{VECTOR}(\text{VECTOR}(1/(\text{DIMENSION}(M_z')-1)*M_z_{\downarrow k}*M_z_{\downarrow l}/(\sigma_{i\downarrow k}*\sigma_{i\downarrow l}),k,1,16),l,1,16)$$

Тут коефіцієнти взаємної коваріації діляться на добуток відповідних середньоквадратичних відхилень. Результуюча матриця симетрична з одиницями на діагоналі.

10.4.3. Обчислення власних значень кореляційної матриці

Більша розмірність кореляційної матриці, навіть із розміром 16×16 , ускладнює вирішення повної проблеми власних значень на обчислювальних машинах з обмеженою розрядною сіткою.

У пакеті символьної математики DERIVE цю задачу можна розв'язати практично з абсолютною точністю, якщо для виконання складених операторів використовувати режим символьних перетворень. У цьому режимі всі проміжні числові операції виконуються з використанням арифметики цілих чисел й арифметики цілих дробів з необмеженою кількістю розрядів. Операція ділення використовується тільки на кінцевому етапі після переходу в режим наближених обчислень із обмеженою розрядною сіткою. Правда режим точних обчислень суттєво уповільнює час розв'язання і суттєво поглинає вільну пам'ять. У лабораторній роботі це не відіграє істотної ролі тому, що всі перетворення проводяться в навчальних цілях і з метою демонстрації і розкриття методу.

Враховуючи сказане, власні значення для кореляційної матриці знайдемо з характеристичного рівняння $Y_p(s)$, яке одержимо шляхом розкриття визначника

$$Y_p(s) := \text{DET}(K - s * \text{IDENTITY_MATRIX}(16))$$

Точне виконання цієї функції видасть на екран монітора неймовірно величезне, але точне вираження щодо єдиної змінної s . Перейшовши в режим побудови графіка функції однієї змінної приблизно оцінить найбільші значення власних значень і запишіть їх.

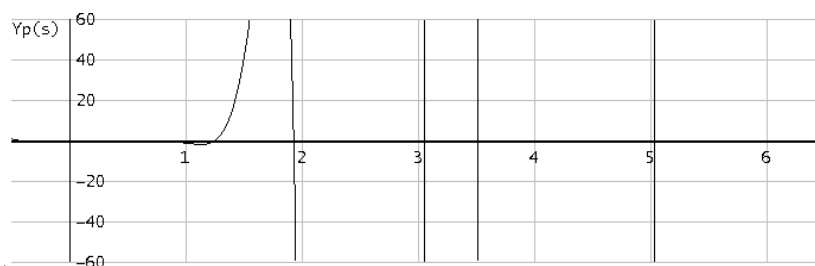


Рисунок 10.1 – Нулі характеристичного рівняння

Після цього для наближеного обчислення дійсних власних значень використовуйте оператор **SOLVE**

$$S := \text{NSOLVE}(Y_p(s), s, \text{Real}).$$

Для сортування і ранжирування власних значень у порядку зменшення їх

величин виконаємо такі вкладені функції

$$Sr:=\text{REVERSE}(\text{SORT}(\text{SOLUTIONS}(S,s))).$$

У цій послідовності внутрішня функція формує вектор із власних значень – сортує його компоненти за зростанням, а зовнішня – подає у зворотному порядку. Наприклад, упорядкований вектор власних значень, виведений на екран монітора з 4 значущими цифрами, буде виглядати так

$$Sr:=[5.033,3.519,3.055,1.925,1.242,0.4569,0.3231,0.3145,0.04301,0.03041, \\ 0.02971,0.01566,0.006735,0.002391,0.0002307,2.037*10^{(-12)}].$$

Для визначення в Sr набору значущих власних значень необхідно обчислити відносні часткові суми компонентів, починаючи з найбільшого

$$\Sigma s:=\text{VECTOR}(\Sigma(Sr\downarrow i,i,1,k)/\Sigma(Sr\downarrow i,i,1,\text{DIMENSION}(Sr)), \\ k,1,\text{DIMENSION}(Sr))$$

Критерієм відбору m значимих власних значень вважатимемо рівень величини відносної часткової суми, що дорівнює, наприклад, 0,95. Останні $[16-(m+1)]$ найменші власні значення можна замінити однаковими константами, величину яких визначимо, наприклад, так

$$\text{const}:=\Sigma(Sr\downarrow k,k,m+1,\text{DIMENSION}(Sr))/\text{DIMENSION}(Sr)$$

Для m відібраних найбільших власних значень обчислимо власні вектори, використовуючи наближену процедуру $\text{APPROX_EIGENVECTOR}(*)$

$$Bs:=\text{APPEND}(\text{VECTOR}(\text{APPROX_EIGENVECTOR}(K,Sr\downarrow k),k,1,m), \\ (\text{DIMENSION}(Sr)-m)*[\text{APPROX_EIGENVECTOR}(K,\text{const})])$$

Сума обчислених власних векторів, помножених на відповідні їм власні значення визначить узагальнений вектор вихідного зображення, що складається із проєкцій ортогональної системи координат, вісями якої служать власні вектори кореляційної матриці

$$[\text{APPEND}(\text{VECTOR}(Sr\downarrow k,k,1,m),[\text{const}])] * Bs.$$

10.5.Звіт та висновки

Звіт має мати коментовані результати всіх етапів одержання розкладання зображень Карунена-Лоєва. Оцінити переваги та недоліки такого стискнення вихідного опису зображення.

Список літератури

1. Калашніков В. І. Основи магнітно-резонансної томографії: навч. посібник/ В.І. Калашніков, І.П. Хавіна – Харків: НТУ “ХПІ”, 2010. – 134 с.
2. Загидуллин Р.Ш. Методические указания к лабораторному практикуму. Основные сведения о системе Micro-Cap 7: метод. указ. – М., 2003. – 33 с.
Метод доступу http://www.bmstu.ru/~rl1/courses/electr/KursRabotaRK6_3k/
3. Разевиг В.Д. Система схемотехнического моделирования Micro-Cap V./ В.Д. Разевиг. – М.: АОТ “ПОЛИТЕХ-4”, 1998. – 273 с.
4. Калашников В.И. Численные методы в среде символьной математики. Программирование и решение задач: учеб. пособ./ В.И. Калашников – Харьков: НТУ “ХПІ”, 2005. – 128 с.
5. Справочник по системе символьной математики DERIVE / Дьяконов В.П. – М.: «СК Пресс», 1998. – 256 с.

Навчальне видання

КАЛАШНІКОВ Володимир Іванович
ХАВІНА Інна Петрівна

ОБРОБКА МАГНІТНО-РЕЗОНАНСНИХ ВІДКЛИКІВ

Лабораторний практикум
з курсу «Спеціалізовані комп'ютерні системи»

Відповідальний за випуск Ф.А. Домнін

Роботу до видання рекомендував проф. М.Й. Заполовський

Редактор В.В. Полянська

План 2009 р. поз. 78/ 35-10

Підписано до друку 06.03.10. Формат 60х84/16. Папір офсетний.
Друк – різнографія. Гарнітура Times New Roman. Ум.-друк. арк. 2,5.
Обл.-вид. арк. 3,0. Наклад 100 прим. Зам. № Ціна договірна

Видавничий центр НТУ “ХПІ”.
Свідоцтво про державну реєстрацію ДК № 116 від 10.07.2004 р.
61002, Харків, вул. Фрунзе, 21.

Друкарня НТУ “ХПІ”, 61002, Харків, вул. Фрунзе, 21.

